Transfer Learning

Qiang Yang
Yu Zhang
Wenyuan Dai
Sinno Jialin Pan

Transfer Learning

Transfer learning deals with how systems can quickly adapt themselves to new situations, new tasks and new environments. It gives machine learning systems the ability to leverage auxiliary data and models to help solve target problems when there is only a small amount of data available in the target domain. This makes such systems more reliable and robust, keeping the machine learning model faced with unforeseeable changes from deviating too much from expected performance. At an enterprise level, transfer learning allows knowledge to be reused so experience gained once can be repeatedly applied to the real world.

This self-contained, comprehensive reference text begins by describing the standard algorithms and then demonstrates how these are used in different transfer learning paradigms and applications. It offers a solid grounding for newcomers as well as new insights for seasoned researchers and developers.

QIANG YANG is the Head of AI at WeBank and a chair professor of computer science and engineering at Hong Kong University of Science and Technology. He is a fellow of the ACM, AAAI, IEEE, IAPR and AAAS, and has served on the AAAI Executive Council and as president of IJCAI. Awards include the 2004/2005 ACM KDDCUP Championship, the ACM SIGKDD Distinguished Service Award and AAAI Innovative AI Applications Award. His books include *Intelligent Planning, Crafting Your Research Future and Constraint-Based Design Recovery for Software Engineering.*

YU ZHANG is an associate professor in the Department of Computer Science and Engineering at Southern University of Science and Technology. He has published about sixty papers in top-tier AI and machine learning conferences and journals. He won the best paper awards at UAI 2010 and PAKDD 2019, and the best student paper award in the 2013 IEEE/WIC/ACM International Conference on Web Intelligence. He was awarded the Young National Distinguished Scholar in China.

WENYUAN DAI is the Founder and CEO of 4Paradigm Co., Ltd. He was a principal architect and senior scientist in Baidu, helping to develop one of China's largest machine learning systems, and a principal scientist in Huawei Noah's Ark Lab. He has published numerous papers in ICML, NIPS, AAAI, KDD and other conferences, primarily on transfer learning and AutoML. He won the ACM-ICPC World Final 2005 and the PKDD best student paper award in 2007, and in 2017 was named *MIT Technology Review* Innovators under 35 in China and Fortune 40 under 40 in China.

SINNO JIALIN PAN is Provost's Chair Associate Professor in the School of Computer Science and Engineering at Nanyang Technological University, Singapore and was formerly Lab Head of Text Analytics with the Data Analytics Department, Institute for Infocomm Research, Singapore. He was named AI 10 to Watch by *IEEE Intelligent Systems* in 2018.

Transfer Learning

QIANG YANG
Hong Kong University of Science and Technology

YU ZHANG
Southern University of Science and Technology

WENYUAN DAI
4Paradigm Co., Ltd.

SINNO JIALIN PAN
Nanyang Technological University

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre,
New Delhi – 110025, India

79 Anson Road, #06–04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107016903
DOI: 10.1017/9781139061773

First published 2020

Printed in the United Kingdom by TJ International, Padstow Cornwall

A catalogue record for this publication is available from the British Library.

ISBN 978-1-107-01690-3 Hardback

Contents

Preface

This book is about the foundations, methods, techniques and applications of transfer learning. Transfer learning deals with how learning systems can quickly adapt themselves to new situations, new tasks and new environments. Transfer learning is a particularly important area of machine learning, which we can understand from several angles. First, the ability to learn from small data seems to be a particularly strong aspect of human intelligence. For example, we observe that babies learn from only a few examples and can quickly and effectively generalize from the few examples to concepts. This ability to learn from small data can be partly explained by the ability of humans to leverage and adapt the previous experience and pretrained models to help solve future target problems. Adaptation is an innate ability of intelligent beings and artificially intelligent agents should certainly be endowed with transfer learning ability.

Second, in machine learning practice, we observe that we are often surrounded with lots of small-sized data sets, which are often isolated and fragmented. Many organizations do not have the ability to collect a huge amount of big data due to a number of constraints that range from resource limitations to organizations interests, and to regulations and concerns for user privacy. This *small-data challenge* is a serious problem faced by many organizations applying AI technology to their problems. Transfer learning is a suitable solution for addressing this challenge because it can leverage many auxiliary data and external models, and adapt them to solve the target problems.

Third, transfer learning can make AI and machine learning systems more reliable and robust. It is often the case that, when building a machine learning model, one cannot foresee all future situations. In machine learning, this problem is often addressed using a technique known as regularization, which leaves room for future changes by limiting the complexity of the models. Transfer learning takes this approach further, by allowing the model to be complex while being prepared for changes when they actually come.

In addition, when facing unforeseeable changes and taking a learned model across domain boundaries, transfer learning still makes sure that the model performance does not deviate from the expected performance too much. In this way,

transfer learning allows knowledge to be reused so experience gained once can be repeatedly applied to the real world. From a software system's perspective, if a system is capable of adapting itself via transfer learning in new domains, it is said to be more robust and more reliable when the external environment changes. Such systems are often preferred in engineering practice.

If we continuously apply transfer learning in our machine learning practice, we can obtain a lifelong machine learning system that can draw knowledge from a succession of problem-solving experience, both in a long period of time and from a large variety of tasks. Transfer learning endows an intelligent system with the lifelong learning ability.

Last, but not least, a transfer learning system can be the backbone of a sound business model in which user privacy is taken into serious consideration, such that a pretrained model can be downloaded and adapted at the edge of a computer network without leaking user data accumulated at the edge or from the cloud. By moving the model one way from a server to a client, the privacy at the client side is effectively protected. In addition, by carefully structuring the transfer learning algorithms, private user information on the cloud side can also be protected.

Like AI in general and machine learning in particular, the concept of transfer learning has gone through decades of evolution. From AI's early years, researchers have considered the ability to transfer one's knowledge as one of the fundamental cornerstones of intelligence. Transfer learning is also given different names and explored under different guises, including learning by analogy, case-based reasoning, knowledge reuse and reengineering, lifelong machine learning, never-ending learning and domain adaption, to name a few. Outside of AI and Computer Science, the concept of transfer learning has also been invented under different terms. In the fields of educational theory and learning psychology, for example, the concept of *transfer of learning* has been an important subject in modeling what constitutes effective learning and teaching for educators; it is believed that the best teaching enables the student to learn "how to learn" and adapt the learned knowledge in future situations. Despite different names, their spirits are all similar: to be able to leverage one's past experience to help make more effective decisions in the future.

The study of transfer learning involves many areas of study in science and engineering, including AI, algorithmic theories, probability and statistics, to name a few. The field is also undergoing rapid changes as interests in AI grow, and many new areas contribute to the field. As the first book of its kind in the area, we hope to use it as a tool to help educate the newcomers of machine learning research and application field, as well as a reference book for seasoned machine learning researchers and application developers to use.

The book is partitioned into two parts. Part I presents the foundations of transfer learning. Chapter 1 gives an overview and introduction to transfer learning. Chapters 2–14 introduce various theoretical and algorithmic aspects of transfer

learning. Part II, which includes Chapters 15–22, covers many application fields of transfer learning. We give concluding remarks in Chapter 23.

The book is an accumulation of hard research work by a group of researchers that spans over a decade, mainly consisting of Professor Qiang Yang's current and former graduate students, postdoctoral researchers and research associates. We have assigned each chapter to one or more students, and then the four main editors either wrote other chapters or went in depth in each chapter to help refine the content, or did both.

The following is a list of these authors.

- Chapter 1: Sinno Jialin Pan and Qiang Yang
- Chapter 2: Xiang Zhang
- Chapter 3: Xu Geng
- Chapter 4: Xueyang Wu
- Chapter 5: Han Tian
- Chapter 6: Ying Wei
- Chapter 7: Yinghua Zhang
- Chapter 8: Bo Liu
- Chapter 9: Yu Zhang
- Chapter 10: Yu Zhang
- Chapter 11: Ben Tan
- Chapter 12: Yu Zhang and Ying Wei
- Chapter 13: Jinliang Deng
- Chapter 14: Lianghao Li and Qiang Yang
- Chapter 15: Xiawei Guo, Yuqiang Chen, Weiwei Tu, and Wenyuan Dai
- Chapter 16: Yinghua Zhang and Weiyan Wang
- Chapter 17: Wenyi Xiao and Zheng Li
- Chapter 18: Kaixiang Mo
- Chapter 19: Weike Pan and Guangneng Hu
- Chapter 20: Qian Xu, Bo Liu and Qiang Yang
- Chapter 21: Vincent W. Zheng and Hao Hu
- Chapter 22: Leye Wang and Yexin Li

Finally, we wish to thank the managerial work of Yutao Deng, who helped keep the schedules and manage team works. To all, our sincere thanks! Without their tremendous effort, the book would have been impossible to complete.

We the editors wish to also thank our colleagues, organizations and collaborators over the years. We thank the support of Hong Kong University of Science and Technology, Hong Kong CERG Fund, Hong Kong Innovation and Technology Fund, the 4Paradigm Corp., Nanyang Technological University Singapore, Webank and many others for their generous support.

Finally, we wish to acknowledge the support of our families, whose patience and encouragement allowed us to finally complete the book.

PART I

FOUNDATIONS OF TRANSFER LEARNING

1
Introduction

1.1 AI, Machine Learning and Transfer Learning

AI was a vision initiated by Alan Turing when he asked the famous question: "Can machines think?" This question has motivated generations of researchers to explore ways to make machines behave intelligently. Throughout recent history, AI has experienced several ups and downs, much of which evolve around the central question of how machines can acquire knowledge from the outside world.

Attempts to make machines think like humans have gone a long way, from force-feeding rule-like knowledge bases to machine learning from data. Machine learning has thus grown from an obscure discipline to a major industrial and societal force in automating decisions that range from online commerce and advertising to education and health care. Machine learning is becoming a general enabling technology for the world due to its strong ability to endow machines with knowledge by letting them learn and adapt through labeled and unlabeled data. Machine learning produces prediction models from data, thus often requiring well-defined data as "teachers" to help tune statistical models. This ability in making accurate predictions of future events are based on observations and understanding of the task domains. The data samples in the training examples are often "labeled," which means that observations and outcomes of predictions in the training data are coupled and correlated. These examples are then used as "teachers" by a machine learning algorithm to "train" a model that can be applied to new data.

One can find many illustrative examples of machine learning in the real world. One example is in the area of face recognition in computer-based image analysis. Suppose that we have obtained a large pool of photos taken indoors. A machine learning system can then use these data to train a model that reports whether a new photo corresponds to a person appearing in the pool. An application of this model would be a gate security system for a building, where a task would be to ascertain whether a visitor is an employee in the organization.

Even though a machine learning model can be made to be of high quality, it can also make mistakes, especially when the model is applied to different scenarios from its training environments. For example, if a new photo is taken from an outdoor environment with different light intensities and levels of noise such as shadows, sunlight from different angels and occlusion by passersby, the recognition capability of the system may dramatically drop. This is because the model trained by the machine learning system is applied to a "different" scenario. This drop in performance shows that models can be outdated and needs updating when new situations occur. It is this need to update or **transfer** models from one scenario to another that lends importance to the topic of the book.

The need for transfer learning is not limited to image understanding. Another example is understanding Twitter text messages by natural language processing (NLP) techniques. Suppose we wish to classify Twitter messages into different user moods such as happy or sad by its content. When one model is built using a collection of Twitter messages and then applied to new data, the performance drops quite dramatically as a different community of people will very likely express their opinions differently. This happens when we have teenagers in one group and grown-ups in another.

As the previous examples demonstrate, a major challenge in practicing machine learning in many applications is that models do not work well in new task domains. The reason why they do not work well may be due to one of several reasons: lack of new training data due to the small data challenge, changes of circumstances and changes of tasks. For example, in a new situation, high-quality training data may be in short supply if not often impossible to obtain for model retraining, as in the case of medical diagnosis and medical imaging data. Machine learning models cannot do well without sufficient training data. Obtaining and labeling new data often takes much effort and resources in a new application domain, which is a major obstacle in realizing AI in the real world. Having well-designed AI systems without the needed training data is like having a sports car without an energy.

This discussion highlights a major roadblock in populating machine learning to the practical world: it would be impossible to collect large quantities of data in every domain before applying machine learning. Here we summarize some of the reasons to develop such a transfer learning methodology:

1) *Many applications only have small data*: the current success of machine learning relies on the availability of a large amount of labeled data. However, high-quality labeled data are often in short supply. Traditional machine learning methods often cannot generalize well to new scenarios, a phenomenon known as overfitting, and fail in many such cases.
2) *Machine learning models need to be robust*: traditional machine learning often makes an assumption that both the training and test data are drawn from the same distribution. However, this assumption is too strong to hold in many

practical scenarios. In many cases, the distribution varies according to time and space, and varies among situations, so we may never have access to new training data to go with the same test distribution. In situations that differ from the training data, the trained models need adaptation before they can be used.

3) *Personalization and specialization are important issues*: it is critical and profitable to offer personalized service for every user according to individual tastes and demands. In many real world applications, we can only collect very little personal data from an individual user. As a result, traditional machine learning methods suffer from the cold start problems when we try to adapt a general model to a specific situation.
4) *User privacy and data security are important issues*: often in our applications we must work with other organizations by leveraging multiple data sets. Often these data sets have different owners and cannot be revealed to each other for privacy or security concerns. When building a model together, it would be desirable for us to extract the "essence" of each data set and adapt them in building a new model. For example, if we can adapt a general model at the "edge" of a network of devices, then the data stored on the device need not to be uploaded to enhance the general model; thus, privacy of the edge device can be ensured.

These objectives for intelligent systems motivated the development of transfer learning. In a nutshell, *transfer learning* refers to the machine learning paradigm in which an algorithm extracts knowledge from one or more application scenarios to help boost the learning performance in a target scenario. Compared to traditional machine learning, which requires large amounts of well-defined training data as the input, transfer learning can be understood as a new learning paradigm, which the rest of the book will cover in detail. Transfer learning is also a motivation to solve the so-called data sparsity and cold start problems in many large-scale and online applications (e.g., labeled user rating data in online recommendation systems may be too few to allow these online systems to build a high-quality recommendation system).

Transfer learning can help promote AI in less-developed application areas, as well as less technically developed geographical areas, even when not much labeled data is available in such areas. For example, suppose we wish to build a book recommendation system in a new online shopping application. Suppose that the book domain is so new that we do not have many transactions recorded in this domain. If we follow the supervised learning methodology in building a prediction model in which we use the insufficient training data in the new domain, we cannot have a credible prediction model on users' next purchase. However, with transfer learning, one can look to a related, well-developed but different domain for help, such as an existing movie recommendation domain. Exploiting transfer learning techniques, one can find the similarity and differences between the book and the movie domains. For example, some authors also turn their books into movies, and movies and books can attract similar user groups. Noticing these similarities can

allow one to focus on adapting the new parts for the book-recommendation task, which allows one to further exploit the underlying similarities between the data sets. Then, book domain classification and user preference learning models can be adapted from those of the movie domain.

Based on the transfer learning methodologies, once we obtain a well-developed model in one domain, we can bring this model to benefit other similar domains. Hence, having an accurate "distance" measure between any task domains is necessary in developing a sound transfer learning methodology. If the distance between two domains is large, then we may not wish to apply transfer learning as the learning might turn out to produce a negative effect. On the other hand, if two domains are "close by," transfer learning can be fruitfully applied.

In machine learning, the distance between domains can often be measured in terms of the features that are used to describe the data. In image analysis, features can be pixels or patches in an image pattern, such as the color or shape. In NLP, features can be words or phrases. Once we know that two domains are close to each other, we can ensure that AI models can be propagated from the well-developed domains to less-developed domains, making the application of AI less data dependent. And this can be a good sign for successful transfer learning applications.

Being able to transfer knowledge from one domain to another allows machine learning systems to extend their range of applicability beyond their original creation. This generalization ability helps make AI more accessible and more robust in many areas where AI talents or resources such as computing power, data and hardware might be scarce. In a way, transfer learning allows the promotion of AI as a more inclusive technology that serves everyone.

To give an intuitive example, we can use an analogy to highlight the key insights behind transfer learning. Consider driving in different countries in the world. In the USA and China, for example, the driver's seat is on the left of the car and drives on the right side of the road. In Britain, the driver sits on the right side of the car, and drives on the left side of the road. For a traveler who is used to driving in the USA to travel to drive in Britain, it is particularly hard to switch. Transfer learning, however, tells us to find the invariant in the two driving domains that is a common feature. On a closer observation, one can find that no matter where one drives, the driver's distance to the center of the road is the closest. Or, conversely, the driver sits farthest from the side of the road. This fact allows human drivers to smoothly "transfer" from one country to another. Thus, the insight behind transfer learning is to find the "invariant" between domains and tasks.

Transfer learning has been studied under different terminologies in AI, such as knowledge reuse and CBR, learning by analogy, domain adaptation, pre-training, fine-tuning, and so on. In the fields of education and learning psychology, transfer of learning has a similar notion as transfer learning in machine learning. In particular, transfer of learning refers to the process in which past experience acquired from previous source tasks can be used to influence future learning and

performance in a target situation (Thorndike and S. Woodworth, 1901). Transfer of learning in the field of education shares a common goal as transfer learning in machine learning in that they both address the process of learning in one context and applying the learning in another. In both areas, the learned knowledge or model is taken to a future target task for use after some adaptation. When one delves into the literature of education theory and learning psychology (Ellis, 1965; Pugh and Bergin, 2006; Schunk, 1965; Cree and Macaulay, 2000), one can find that, despite the fact that transfer learning in machine learning aims to endow machines with the ability to adapt and transfer of learning in education tries to study how humans adapt in education, the processes or algorithms of transfer are similar.

A final note on the benefit of transfer learning is in simulation technology. Often in complex tasks, such as robotics and drug design, for example, it is too expensive to engage real world experiments. In robotics, a mobile robot or an autonomous vehicle needs to collect sufficient training data. For example, there may be many ways in which a car is involved in a car crash but to create car crashes is far too expensive in real life. Instead, researchers often build sophisticated simulators such that a trained model taught in the simulator environment is applied to the real world after adaptation via transfer learning. The transfer learning step is needed to account for many future situations that are not seen in the simulated environment and adapt the simulated prediction models, such as obstacle avoidance models in autonomous cars, to unforeseeable future situations.

1.2 Transfer Learning: A Definition

To start with, we define what "domain," "task" and "transfer learning" mean by following the notations introduced by Pan and Yang (2010). A *domain* $\mathbb{D}$ consists of two components: a feature space $\mathscr{X}$ and a marginal probability distribution $\mathbb{P}^X$, where each input instance $\mathbf{x} \in \mathscr{X}$. In general, if two domains are different, then they may have different feature spaces or different marginal probability distributions. Given a specific domain, $\mathbb{D} = \{\mathscr{X}, \mathbb{P}^X\}$, a *task* $\mathbb{T}$ consists of two components: a label space $\mathscr{Y}$ and a function $f(\cdot)$ (denoted by $\mathbb{T} = \{\mathscr{Y}, f(\cdot)\}$). The function $f(\cdot)$ is a predictive function that can be used to make predictions on unseen instances $\{\mathbf{x}^*\}$s. From a probabilistic viewpoint, $f(\mathbf{x})$ can be written as $P(y|\mathbf{x})$. In classification, labels can be binary, that is, $\mathscr{Y} = \{-1, +1\}$, or discrete values, that is, multiple classes. In regression, labels are of continuous values.

For simplicity, we now focus on the case where there are one source domain $\mathbb{D}_s$ and one target domain $\mathbb{D}_t$. The two-domain scenario is by far the most popular of the research works in the literature. In particular, we denote by $\mathscr{D}_s = \{(\mathbf{x}_{s_i}, y_{s_i})\}_{i=1}^{n_s}$ the *source domain labeled data*, where $\mathbf{x}_{s_i} \in \mathscr{X}_s$ is the data instance and $y_{s_i} \in \mathscr{Y}_s$ is the corresponding class label. Similarly, we denote by $\mathscr{D}_t = \{(\mathbf{x}_{t_i}, y_{t_i})\}_{i=1}^{n_t}$ the target domain labeled data, where the input $\mathbf{x}_{t_i}$ is in $\mathscr{X}_t$ and $y_{t_i} \in \mathscr{Y}_t$ is the corresponding

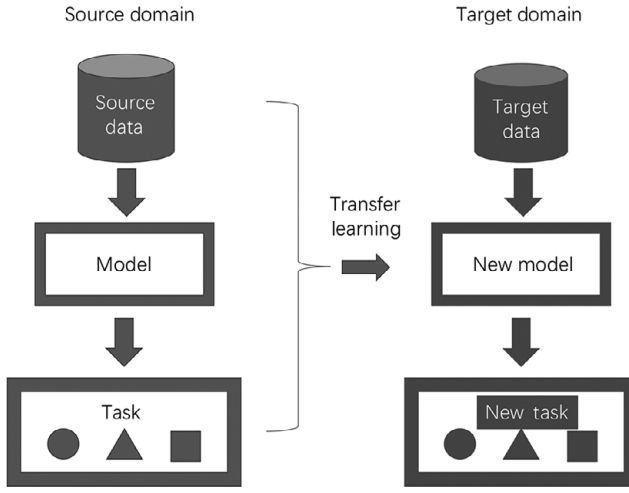

Figure 1.1 An illustration of a transfer learning process

output. In most cases, $0 \leq n_t \ll n_s$. Based on these notations, transfer learning can be defined as follows (Pan and Yang, 2010).

Definition 1.1 (transfer learning) Given a source domain $\mathbb{D}_s$ and learning task $\mathbb{T}_s$, a target domain $\mathbb{D}_t$ and learning task $\mathbb{T}_t$, *transfer learning* aims to help improve the learning of the target predictive function $f_t(\cdot)$ for the target domain using the knowledge in $\mathbb{D}_s$ and $\mathbb{T}_s$, where $\mathbb{D}_s \neq \mathbb{D}_t$ or $\mathbb{T}_s \neq \mathbb{T}_t$.

A transfer learning process is illustrated in Figure 1.1. The process on the left corresponds to a traditional machine learning process. The process on the right corresponds to a transfer learning process. As we can see, transfer learning makes use of not only the data in the target task domain as input to the learning algorithm, but also any of the learning process in the source domain, including the training data, models and task description. This figure shows a key concept of transfer learning: it counters the lack of training data problem in the target domain with more knowledge gained from the source domain.

As a domain contains two components, $\mathbb{D} = \{\mathscr{X}, \mathbb{P}^X\}$, the condition $\mathbb{D}_s \neq \mathbb{D}_t$ implies that either $\mathscr{X}_s \neq \mathscr{X}_t$ or $\mathbb{P}^{X_s} \neq \mathbb{P}^{X_T}$. Similarly, as a task is defined as a pair of components $\mathbb{T} = \{\mathscr{Y}, \mathbb{P}^{Y|X}\}$, the condition $\mathbb{T}_s \neq \mathbb{T}_t$ implies that either $\mathscr{Y}_s \neq \mathscr{Y}_t$ or $\mathbb{P}^{Y_s|X_s} \neq \mathbb{P}^{Y_t|X_t}$. When the target domain and the source domain are the same, that is, $\mathbb{D}_s = \mathbb{D}_t$, and their learning tasks are the same, that is, $\mathbb{T}_s = \mathbb{T}_t$, the learning problem becomes a traditional machine learning problem.

Based on this definition, we can formulate different ways to categorize existing transfer learning studies into different settings. For instance, based on the homogeneity of the feature spaces and/or label spaces, we can categorize transfer

learning into two settings: (1) homogeneous transfer learning and (2) heterogeneous transfer learning, whose definitions are described as follows (Pan, 2014).[1]

Definition 1.2 (homogeneous transfer learning) Given a source domain $\mathbb{D}_s$ and a learning task $\mathbb{T}_s$, a target domain $\mathbb{D}_t$ and a learning task $\mathbb{T}_t$, *homogeneous transfer learning* aims to help improve the learning of the target predictive function $f_t(\cdot)$ for $\mathbb{D}_t$ using the knowledge in $\mathbb{D}_s$ and $\mathbb{T}_s$, where $\mathscr{X}_s \bigcap \mathscr{X}_t \neq \emptyset$ and $\mathscr{Y}_s = \mathscr{Y}_t$, but $\mathbb{P}^{X_s} \neq \mathbb{P}^{X_t}$ or $\mathbb{P}^{Y_s|X_s} \neq \mathbb{P}^{Y_t|X_t}$.

Definition 1.3 (heterogeneous transfer learning) Given a source domain $\mathbb{D}_s$ and a learning task $\mathbb{T}_s$, a target domain $\mathbb{D}_t$ and a learning task $\mathbb{T}_t$, *heterogeneous transfer learning* aims to help improve the learning of the target predictive function $f_t(\cdot)$ for $\mathbb{D}_t$ using the knowledge in $\mathbb{D}_s$ and $\mathbb{T}_s$, where $\mathscr{X}_s \bigcap \mathscr{X}_t = \emptyset$ or $\mathscr{Y}_s \neq \mathscr{Y}_t$.

Besides using the homogeneity of the feature spaces and label spaces, we can also categorize existing transfer learning studies into the following three settings by considering whether labeled data and unlabeled data are available in the target domain: supervised transfer learning, semi-supervised transfer learning and unsupervised transfer learning. In supervised transfer learning, only a few labeled data are available in the target domain for training, and we do not use the unlabeled data for training. For unsupervised transfer learning, there are only unlabeled data available in the target domain. In semi-supervised transfer learning, sufficient unlabeled data and a few labeled data are assumed to be available in the target domain.

To design a transfer learning algorithm, we need to consider the following three main research issues: (1) when to transfer, (2) what to transfer and (3) how to transfer.

When to transfer asks in which situations transferring skills should be done. Likewise, we are interested in knowing in which situations knowledge should **not** be transferred. In some situations, when the source domain and the target domain are not related to each other, brute-force transfer may be unsuccessful. In the worst case, it may even hurt the performance of learning in the target domain, a situation which is often referred to as *negative transfer*. Most of current studies on transfer learning focus on "what to transfer" and "how to transfer," by implicitly assuming that the source domain and the target domain are related to each other. However, how to avoid negative transfer is an important open issue that is attracting more and more attentions.

What to transfer determines which part of knowledge can be transferred across domains or tasks. Some knowledge is specific for individual domains or tasks, and some knowledge may be common between different domains such that they may help improve performance for the target domain or task. Note that the term

[1] In the rest of book, without explicit specification, the term "transfer learning" denotes homogeneous transfer learning.

"knowledge" is very general. Thus, in practice, it needs to be specified based on different context.

How to transfer specifies the form that a transfer learning method takes. Different answers to the question of "how to transfer" give a categorization for transfer learning algorithms:

(1) instance-based algorithms, where the knowledge transferred corresponds to the weights attached to source instances;
(2) feature-based algorithms, where the knowledge transferred corresponds to the subspace spanned by the features in the source and target domains;
(3) model-based algorithms, where the knowledge to be transferred is embedded in part of the source domain models and
(4) relation-based algorithms, where the knowledge to be transferred corresponds to rules specifying the relations between the entities in the source domain.

Each of these types of transfer learning corresponds to an emphasis on which part of the knowledge is being considered as a vehicle to facilitate the knowledge transfer. Specifically, a common motivation behind **instance-based transfer learning approaches** is that, although the source domain labeled data cannot be reused directly due to the domain difference, part of them can be reused for the target domain after reweighting or resampling. In this way, the source-domain labeled instances with large weights can be considered as "knowledge" to be transferred across domains. An implicit assumption behind the instance-based approaches is that the source domain and the target domain have a lot of overlapping features, which means that the domains share the same or similar support.

However, in many real world applications, only a portion of the feature spaces from the source and target domains overlap, which means that many features cannot be directly used as bridges for the knowledge transfer. As a result, some instance-based methods may fail to work effectively for knowledge transfer. **Feature-based transfer learning approaches** are more promising in this case. A common idea behind feature-based approaches is to learn a "good" feature representation for both the source domain and the target domain such that, by projecting data onto the new representation, the source domain labeled data can be reused to train a precise classifier for the target domain. In this way, the knowledge to be transferred across domains can be considered as the learned feature representation.

Model-based transfer learning approaches assume the source domain and the target domain share some parameters or hyperparameters of the learning models. A motivation of model-based approaches is that a well-trained source model has captured a lot of useful structure, which is general and can be transferred to learn a more precise target model. In this way, the knowledge to be transferred is the domain-invariant structure of the model parameters. A recently widely used pretraining technique for transfer learning based on deep learning is indeed a model-based approach. Specifically, the idea of pretraining is to first train a deep

learning model using sufficient source data, which could be quite different from the target data. After the deep model is trained, a few target labeled data are used to fine-tune part of the parameters of the pretrained deep model, for example, to fine-tune parameters of several layers while fixing parameters of other layers.

Different from the three aforementioned categories of approaches, **relation-based transfer learning approaches** assume that some relationships between objects (i.e., instances) are similar across domains or tasks. Once these common relationships are extracted, then they can be used as knowledge for transfer learning. Note that, in this category, data in the source domain and the target domain are not required to be independent and identically distributed as the other three categories.

1.3 Relationship to Existing Machine Learning Paradigms

Transfer learning and machine learning are closely related. On one hand, the aim of transfer learning encompasses that of machine learning in that its key ingredient is "generalization." In other words, it explores how to develop general and robust machine learning models that can apply to not only the training data, but also unanticipated future data. Therefore, all machine learning models should have the ability to conduct transfer learning. On the other hand, transfer learning differs from other branches of machine learning in that transfer learning aims to generalize commonalities across different tasks or domains, which are "sets" of instances, while machine learning focuses on generalize commonalities across "instances." This difference makes the design of the learning algorithms quite different.

Specifically, machine learning algorithms such as semi-supervised learning, active learning and transfer learning can be used to partially address the labeled data sparsity issue for a target domain, but they have different assumptions. Semi-supervised learning aims to address the labeled data sparsity problem in the same domain by making use of a large amount of unlabeled data to discover an intrinsic data structure to effectively propagate label information. Common assumptions behind semi-supervised learning techniques are (1) the underlying intrinsic data structure is very useful to learn a precise model even without sufficient labeled data and (2) the training data, including labeled and unlabeled, and the unseen test data are still represented in the same feature space and drawn from the same data distribution.

Instead of exploring unlabeled data to train a precise model, active learning, which is another branch in machine learning for reducing the annotation effort of supervised learning, tries to design an active learner to pose queries, usually in the form of unlabeled data instances to be labeled by an oracle (e.g., a human annotator). The key motivation behind active learning is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed

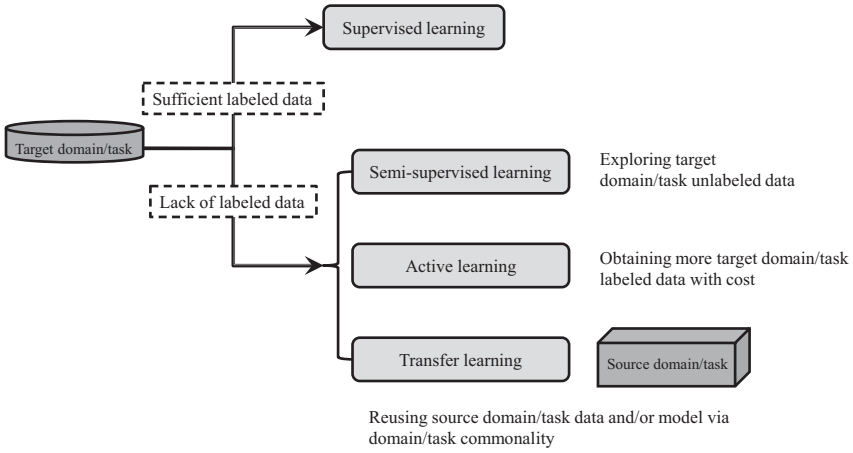

Figure 1.2 Relationship of transfer learning to other learning paradigms

to choose the data from which it learns. However, active learning assumes that there is a budget for the active learner to pose queries in the domain of interest. In some real world applications, the budget may be quite limited, which means that the labeled data queried by active learning may not be sufficient enough to learn an accurate classifier in the domain of interest.

Transfer learning, in contrast, allows the domains, tasks and distributions used in the training phase and the testing phase to be different. The main idea behind transfer learning is to borrow labeled data or extract knowledge from some related domains to help a machine learning algorithm to achieve greater performance in the domain of interest. Thus, transfer learning can be referred to as a different strategy for learning models with minimal human supervision, compared to semi-supervised and active learning.

One of the most related learning paradigms to transfer learning is multi-task learning. Although both transfer learning and multitask learning aim to generalize commonality across tasks, transfer learning is focused on learning on a target task, where some source task(s) is(are) used as auxiliary information, while multitask learning aims to learn a set of target tasks jointly to improve the generalization performance of each learning task without any source or auxiliary tasks. As most existing multitask learning methods consider all tasks to have the same importance, while transfer learning only takes the performance of the target task into consideration, some detailed designs of the learning algorithms are different. However, most existing multitask learning algorithms can be adapted to the transfer learning setting.

We summarize the relationships between transfer learning and other machine learning paradigms in Figure 1.2, and the difference between transfer learning and multitask learning in Figure 1.3.

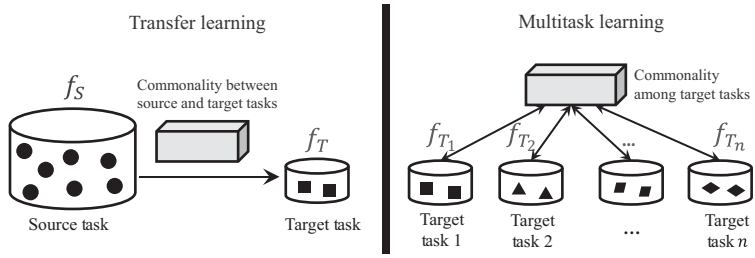

Figure 1.3 Relationship between transfer learning and multitask learning

1.4 Fundamental Research Issues in Transfer Learning

As we mentioned earlier, there are three research issues in transfer learning, namely, "what to transfer," "how to transfer" and "when to transfer." As the objective of transfer learning is to transfer knowledge across different domains, the first question is to ask what knowledge across domains can be transferred to boost the generalization performance of the target domain, which is referred to as the "what to transfer" issue. After identifying what knowledge to be transferred, a follow-up question is how to encode the knowledge into a learning algorithm to transfer, which corresponds to the "how to transfer" issue. The "when to transfer" issue is to ask in which situations transfer learning should be performed or can be performed safely. A fundamental research question behind these three issues is how to measure the "distance" between any pair of domains or tasks. With the distance measure between domains or tasks, one can identify what common knowledge between tasks can be used to reduce distance between domains or tasks, that is, what to transfer, and figure out how to reduce the distance between domains or tasks based on the identified common knowledge, that is, how to transfer. Moreover, with the distance measure between domains or tasks, one can logically decide "when to transfer": if the distance is very large, it is advised not to conduct transfer learning. Otherwise, it is "safe" to do so.

A subsequent question is thus: what form should such a notion of distance measure be in? Traditionally, there are various types of statistical measures for the distance between any two probability distributions. Typical measures among them include Kulback–Leibler divergence, A-distance (which measures the domain separation) and Maximum Mean Discrepancy (MMD), to name a few. Recall that a domain contains two components: a feature space and a marginal probability distribution, and a task also contains two components: a label space and a conditional probability distribution. Therefore, existing statistical measures for the distance between probability distributions could be used to measure the distance between domains or tasks by assuming the source domain (task) and the target domain (task) share the same feature (label) space. However, there are some limitations on using statistical distance measures for transfer learning. First, researchers have found that these general distribution-based distance measures are

often too coarse to serve the purpose well in measuring the distance in the transferrability between two domains or tasks. Second, if the domains have different feature spaces and/or label spaces, one has to first project the data onto the same feature and/or label space, and then apply the statistical distance measures as a follow-up step. Therefore, more research needs to be done on a general notion of distances between two domains or tasks.

1.5 Applications of Transfer Learning

1.5.1 Image Understanding

Many image understanding tasks from object recognition to activity recognition have been considered. Typically, these computer vision tasks require a lot of labeled data to train a model, such as using the well-known ImageNet data set. However, when computer vision situations slightly change, such as changing from indoors to outdoors and from still cameras to moving cameras, the model needs to be adapted to account for new situations. Transfer learning is an often used technique to solve these adaptation problems.

In image analysis, many recent works combined deep learning architecture with transfer learning. For example, Long et al. (2015) explore a deep learning architecture in which domain distances are minimized between the source and target domains. In a paper published by Facebook (Mahajan et al., 2018), Mahajan et al. apply transfer learning to image classification. The approach involves first training a deep learning model based on a very large image data set. This pretrained model is then fine-tuned on specific tasks in a target domain, which involves relatively small amounts of labeled data. The model is a deep convolutional network trained for the task of classification based on hashtags assigned to billions of social media images, and the target tasks are object recognition or image classification. Their analysis shows that it is important to both increase the size of the pretraining data set as well as to select a closely related label space between source and target tasks. This observation suggests that transfer learning requires the design of "label-space engineering" approaches to match source and target learning tasks. Their work also suggests that improvements on target tasks may be obtained by increasing source model complexity and data set sizes.

Transfer learning also allows image analysis to play an important role in applications with a large societal impact. In the work by Xie et al. (2016), authors from Stanford University Earth Sciences apply transfer learning to predict poverty levels on earth based on satellite images. First, they used daytime images to predict the nighttime light images. The resulting model is then transferred to predicting poverty. This results in a very accurate prediction model that required much less human labeling effort to build compared to traditional survey-based methods.

1.5.2 Bioinformatics and Bio-imaging

In biology, many experiments are costly and data are very few. Examples include bio-imaging when doctors try to use computers to discover potential diseases, and when software models are used to scan complex DNA and protein sequences for patterns to point to a particular illness or cure. Transfer learning has been increasingly used to help leverage the knowledge from one domain to another to address the difficulty that labeled data in biology is costly to obtain. For example, Xu and Yang (2011) give an early survey of transfer learning and multitask learning in bioinformatics applications, and Xu et al. (2011) present a transfer learning process to identify protein cellular structures in a target domain where the labeled data is in short supply. In biomedical image analysis, a difficult problem is to collect enough training data to train a model for identifying image patterns that designate illnesses such as cancer. Such identification requires large amounts of training data. However, these data are often very expensive to obtain as they require costly human experts to label. Furthermore, the data for pretrained models and future models are often from different distributions. These problems inspire many research works to apply transfer learning to adapt the pretrained model in new tasks. For example, in the work by Shin et al. (2016), a pretrained model based on ImageNet data is used as the source domain model, which is then transferred for use in a medical image domain for thoraco-abdominal lymph node detection and interstitial lung disease classification, with great success.

1.5.3 Recommender Systems and Collaborative Filtering

It is often the case that an online product recommendation system is difficult to set up due to the cold start problem. This problem can be alleviated if we discover similarities between domains and adapt a recommendation model from a mature domain to the new domain. This often saves time and resources that make an otherwise impossible task successful. For example, Li et al. (2009b) and Pan et al. (2010b) give early accounts of applying transfer learning for online recommendation. In their applications, cross-domain recommendation systems transfer user preference models from an existing domain (say, a book recommendation domain) to a new domain (say, a movie recommendation domain). The scenario corresponds to the business case where an online commerce site opens a new line of business and wishes to quickly deploy a recommendation model for the operation in the new business line. In doing so, it must overcome the problem of a lack of transaction data in the new business line. Another line of work is in integrating reinforcement learning and recommendation systems to allow the items that are recommended to be both accurate according to past history of a user and potentially diverse to enrich users' interests. As an example, Liu et al. (2018) present a bandit algorithm that balances between recommendation accuracy and topic

diversity, to allow a system to explore new topics as well as cater to users' recent choices. Relating to transfer learning, the work shows that the recommendation strategy in balancing exploration and exploitation can indeed be transferred between domains.

1.5.4 Robotics and Autonomous Cars

In designing robotics and autonomous cars, learning from simulations is a particularly useful approach. These are examples of hardware interactions, where it is costly to gather labeled data for training reinforcement learning and supervised learning models. Taylor and Stone (2007) described how transfer learning helps by allowing researchers to build a simulated model in a more or less ideal domain, the source domain, and then learn a policy to deal with the anticipated events in a target domain. The target domain model can handle more cases in the real world to further handle more unanticipated and noisy data. When the models adapt well, much labor and many resources can be saved from retraining the target domain model. In the work by Tai et al. (2017), a mapless motion planner was designed based on a ten-dimensional sparse range findings and trained in an end-to-end deep reinforcement learning algorithm. Then the learned planner is transferred to the real world by generalizing via real world samples.

1.5.5 NLP and Text Mining

Text mining is a good application for transfer learning algorithms. Text mining aims to discover useful structural knowledge from text and applies to other domains. Among all the problems in text mining, text classification aims to label new text documents with different class tags. A typical text classification problem is *sentiment classification*. On the Web, there are enormous user-generated contents at online sites such as online forums, blogs, social networks and so on. It is very important to be able to summarize opinions of consumers on products and services. Sentiment classification addresses this problem by classifying the reviews into positive and negative categories. However, on different domains, such as different types of products, different types of online sites and different sectors of business, users may express their opinions using different words. As a result, a sentiment classifier trained on one domain may perform poorly on other domains. In this case, transfer learning can help adapt a well-trained sentiment classifier across different domains.

Recently, work on pretraining gained new insights into the nature of transfer learning. Devlin et al. (2018) highlight one successful condition for transfer learning applications: having a sufficient amount of source domain training data. For example, Google's NLP system BERT (Bidirectional Encoder Representations from Transformers) applies transfer learning to a number of NLP tasks, showing that transfer learning with a powerful pretained model can solve a variety of tradition-

ally difficult problems such as question answering problems (Devlin et al., 2018). It has accomplished surprising results by leading in many tasks in the open competition SQuAD 2.0 (Rajpurkar et al., 2016). The source domain consists of an extremely large collection of natural language text corpus, with which BERT trained a model that is based on the bidirectional transformers based on the attention mechanism. The pertained model is capable of making a variety of predictions in a language model more accurate than before, and the predictive power increases with increasing amounts of training data in the source domain. Then, the BERT model is applied to a specific task in a target domain by adding additional small layers to the source model in such tasks as Next Sentence classification, Question Answering and Named Entity Recognition (NER). The transfer learning approach corresponds to model-based transfer, where most hyperparameters stay the same but a selected few hyperparameters can be adapted with the new data in the target domain.

1.6 Historical Notes

Many human learning activities follow the style of transfer learning. We observe that people often apply the knowledge gained from previous learning tasks to help learn a new task. For example, a baby can be observed to first learn how to recognize its parents before using this knowledge to help it learn how to recognize other people.

Transfer learning has deep roots in AI, psychology, educational theory and cognitive science. In AI, there have been many forms of transfer learning. Learning by analogy is one of the fundamental insights of AI. Humans can draw on the past experience to solve current problems very well. In AI, there have been several early works on analogical reasoning such as dynamic memory (Schank, 1983). Using analogy in problem solving, Carbonell (1981) and 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 et al. (1998) have argued for high-level structural similarity as a basis of analogical reasoning. Holyoak and Thagard (1989) have developed a computational theory of analogical reasoning using this strategy, when abstraction rules that allow the two instances to be mapped to a unified representation are given as input.

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 counterarguments. PRODIGY (Carbonell et al., 1991) uses a collection of previous problem-solving cases as a case base, and retrieves the most similar cases for adaptation. Most operational systems of analogical reasoning such as CBR systems (Kolodner, 1993) have relied on an assumption that the past instances and the new target problem are in the same representational space.

Table 1.1 *Notations*

$\mathscr{D}$	A data set
$\mathscr{X}$	A feature space
$\mathscr{H}$	A hypothesis space
$\mathbb{P}$	A probability distribution
$\mathbb{E}_{\mathbb{P}}[\cdot]$	Expectation with respect to distribution $\mathbb{P}$
$\mathrm{tr}(\mathbf{A})$	Trace of matrix $\mathbf{A}$
min	Minimization
max	Maximization
$\mathbf{I}_n$	An $n \times n$ identity matrix
$\mathbf{I}$	An identity matrix with the size depending on the context
$\mathbf{0}$	A zero vector or matrix with the size depending on the context
$\mathbf{1}$	A vector or matrix of all ones with the size depending on the context
$\|\cdot\|_p$	The ℓ_p norm of a vector where $0 \leq p \leq \infty$
$\|\cdot\|_1$	The ℓ_1 norm of a vector or matrix
$\|\cdot\|_F$	The Frobenius norm of a matrix
$\|\cdot\|_{S(p)}$	The Schatten p-norm norm of a matrix
$\mu_i(\cdot)$	The i-th largest eigenvalue or singular value of a matrix
$N(\mu,\sigma)$	A univariate or multivariate normal distribution with mean μ and variance σ
$\|\mathbf{A}\|_{p,q}$	The $\ell_{p,q}$ norm of a matrix, that is, $\|\mathbf{A}\|_{p,q} = \big\|\big(\|\mathbf{a}_1\|_p, \ldots, \|\mathbf{a}_n\|_p\big)\big\|_q$ where $\mathbf{a}_i$ is the ith row of $\mathbf{A}$.
$\mathbf{A}^{-1}$	The inverse of a nonsingular matrix $\mathbf{A}$
$\mathbf{A}^{+}$	The inverse of a nonsingular matrix $\mathbf{A}$ or the pseduo-inverse of a singular matrix

There have been some surveys on transfer learning in machine learning literature. Pan and Yang (2010) and Taylor and Stone (2009) give early surveys of the work on transfer learning, where the former focused on machine learning in classification and regression areas and the latter on reinforcement learning approaches. This book aims to give a comprehensive survey that cover both these areas, as well as the more recent advances of transfer learning with deep learning.

1.7 About This Book

This book mainly consists of two parts. The first part is to introduce the foundation of transfer learning in terms of representative methodologies and theoretical studies. The second part is to discuss some advanced topics in transfer learning and show some successful applications of transfer learning. The notations used in this book are summarized in Table 1.1.

The book is the effort of years of original research and survey of the research field by many former and current students of Professor Qiang Yang at Hong Kong University of Science and Technology and several other organizations. In chronological order of chapters, the composition of the book is outlined as follows:

Chapter 2 covers instance-based transfer learning. One of the most straightforward transfer learning methods is to identify instances or samples from the source domains and assign them weights. Then, these instances with sufficiently high weights are transferred to the target domain to help train a better machine learning model. In doing so, it is important to transfer only those instances that can contribute to the learning in the target domain and at the same time avoid "negative transfer." The instance-based

transfer learning methods can also be useful when multiple source domains exist.

Chapter 3 covers feature-based transfer learning. Features constitute a major element of machine learning. They can be straightforward attributes in the input data, such as pixels in images or words and phrases in a text document, or they can be composite features composed by certain nonlinear transformations of input features. Together these features comprise a high-dimensional feature space. Feature-based transfer is to identify common subspaces of features between source and target domains, and allow transfer to happen in these subspaces. This style of transfer learning is particularly useful when no clear instances can be directly transferred, but some common "style" of learning can be transferred.

Chapter 4 discusses model-based transfer learning. Model-based transfer is when parts of a learning model can be transferred to a target domain from a source domain, where the learning in the target domain can be "fine-tuned" based on the transferred model. Model-based transfer learning is particularly useful when one has a fairly complete collection of data in a source domain, and the model in the source domain can be made very powerful in terms of coverage. Then learning in a target domain corresponds to adapting the general model from the source domain to a specific model in a target domain on the "edge" of a network of domains.

Chapter 5 explores relation-based transfer learning. This chapter is particularly useful when knowledge is coded in terms of a knowledge graph or in relational logic form. When some dictionary of translation can be instituted, and when knowledge exists in the form of some encoded rules, this type of transfer learning can be particularly useful.

Chapter 6 presents heterogeneous transfer learning. Sometimes, when we deal with transfer learning, the target domain may have a completely different feature representation from that of the source domain. For example, we may have collected labeled data about images, but the target task is to classify text documents. If there is some relationship between the images and the text documents, transfer learning can still happen at the semantic level, where the semantics of the common knowledge between the source and the target domains can be extracted as a "bridge" to enable the knowledge transfer.

Chapter 7 discusses adversarial transfer learning. Machine learning, especially deep learning, can be designed to generate data and at the same time classify data. This dual relationship in machine learning can be exploited to mimic the power of imitation and creation in humans. This learning process can be modeled as a game between multiple models, and is called adversarial learning. Adversarial learning can be very useful in empowering a transfer learning process, which is the subject of this chapter.

Chapter 8 discusses the use of transfer learning in reinforcement learning. Reinforcement learning allows rewards to be delayed, and introduces the concept of actions and states in a learning system. Learning a policy in a reinforcement learning problem requires a huge amount of training data, which is time consuming to prepare. Transfer learning alleviates this pain and is promising when the source and target domains and tasks can be closely aligned.

Chapter 9 discusses multitask learning. So far, transfer learning has been discussed along a time line: a source domain and a model have been well prepared before transfer learning can happen to a target domain in a later time point. Multitask learning aims to learn at the same time point, by allowing several tasks to benefit with common knowledge for each other. This is the style of learning when a student takes several courses in the same semester, when the student finds that some common contents or learning methodology can be commonly shared between the courses.

Chapter 10 discusses transfer learning theory. Learning theory tells the general capability of a learning system, by relating the number of samples with the generalization error bounds of a particular algorithm. This line of work generally follows the methodology of probably approximately correct learning, or PAC learning. When the bound is tight, the error bound can also be used to design new algorithms. The transfer learning theory, when properly done, can help give assurances for a learning system's capability.

Chapter 11 surveys transitive transfer learning. Transfer learning so far has been discussed in a source to target domain transfer model. When the source and target domains are "far" from each other, there is no directly relation between the two, transfer cannot directly happen between the two domains. Even though this poses difficulty for transfer learning, there are still opportunities for transfer learning when we can find some intermediate domains as "stepping-stones" for knowledge to "hop over" to target domains. For example, this might happen when we consider a student entering a university taking a calculus class; through several semesters' of knowledge transfer, they eventually they take some advanced physics or computing classes.

Chapter 12 presents learning to transfer as a way to achieve automated transfer learning. Just like a typical machine learning system, the engineering process can be very tedious, as there may be many parameters to tune. As a result, researchers introduced the concept of automatic machine learning (AutoML) to automate the parameter tuning process through automatic optimization. Likewise, transfer learning requires many engineering efforts, and, when sufficient transfer learning experience is gained, this experience can in turn become the training data for building a parameter-tuning model for automatic transfer learning (AutoTL).

Chapter 13 presents few-shot learning. Few-shot learning is when models have been built well enough in a source domain, there may be cases where only few training data, or even no training data, are required in the target domain before a target domain model is well trained.

Chapter 14 discusses lifelong machine learning. When transfer learning is engaged continuously along a time line, the system can draw knowledge from all previous experience in a lifelong manner. A challenge is to decide how to store the previous knowledge and how to select the previous experience to reuse when solving the next task in life.

Chapter 15 discusses privacy-preserving transfer learning. When transfer learning happens between two organizations, we wish to protect the sensitive and private information about users and the confidential data in the source domain. We wish to do this while transferring the knowledge itself. Thus, care should be taken not to allow the target domain to reverse engineer the sensitive data when transfer learning is applied. In this chapter, we discuss how differential privacy is integrated with transfer learning to protect the user privacy and ensure data confidentiality.

Chapter 16 discusses applications of transfer learning in computer vision, which is one of the most extensive application fields of transfer learning. We survey the work in this area, paying special attention to medical imaging and transfer learning.

Chapter 17 discusses applications of transfer learning in NLP. NLP is one of the main application areas of transfer learning, which requires special attention due to the language specific nature of NLP.

Chapter 18 discusses applications of transfer learning in dialogue systems. We particularly separated dialogue systems out of the general survey on NLP in the previous chapter because this is an increasingly important application area not only in its own right, but also as a human–computer interaction medium that will grow in the years to come.

Chapter 19 presents applications of transfer learning in recommendation systems. Recommendation systems is a machine learning technique and, at the same time, an important application area of machine learning. Transfer learning is particularly important in recommendation systems because this domain constantly suffers from the so-called "cold start" problem and data sparsity problem where not enough data and knowledge have been gained in a newly started area. Transfer learning has proven to be very useful in alleviating these problems.

Chapter 20 discusses applications of transfer learning in bioinformatics and bioimaging. Biological data are increasingly accumulated with advancement of genetic and biomedical technology. This gives application opportunities to machine learning. However, this is a domain where collecting high-quality samples is extremely difficult, expensive and time consuming. Thus, transfer learning can be very useful, especially when the ge-

netics domain is full of data of very high dimensionality and low sample sizes. We give an overview of works in this area.

Chapter 21 presents applications of transfer learning in activity recognition based on sensors. Activity recognition refers to finding people's activities from sensor readings, which can be very useful for assisted living, security and a wide range of other applications. A challenge in this domain is the lack of labeled data, and this challenge is particularly fit for transfer learning to address.

Chapter 22 discusses applications of transfer learning in urban computing. There are many machine learning problems to address in urban computing, ranging from traffic prediction to pollution forecast. When data has been collected in one city, the model can be transferred to a newly considered city via transfer learning, especially when there is not sufficient high-quality data in these new cities.

Chapter 23 gives a summary of the whole book with an outlook for future works.

2

Instance-Based Transfer Learning

2.1 Introduction

Intuitively, instance-based transfer learning approaches aim to reuse labeled data from the source domain help to train a more precise model for a target learning task. If the source domain and the target domain are quite similar, we can directly merge the source domain data into the target domain. Then it becomes a standard machine learning problem in a single domain. However, in many cases, this "direct adoption" strategy of source domain instances cannot help to solve the target task.

A common motivation behind instance-based transfer learning approaches is that some source domain labeled data are still useful for learning a precise model for the target domain while some are useless or even may hurt the performance of the target model if used. We can use the bias-variance analysis to understand this motivation. When the target domain data set is small, the model may have a high variance level and thus the model's generalization error is large. By adding a part of the source domain data as an auxiliary data set, the model's variance can potentially be reduced. However, if the data distributions of the two domains are very different, the new learning model may have a high bias. Therefore, if we can single out those source domain instances that follow a similar distribution as those in the target domain, we can reuse them and have both the variance and bias of the target learning model reduced.

Briefly, there are two key issues to resolve in using instance-based transfer learning. The first issue is how to single out the source domain-labeled instances that are similar to the target domain ones, because these instances are useful to train the target domain model. The second issue is how to utilize the identified "similar" source domain-labeled instances in an algorithm to learn a more accurate target domain learning model.

Recall that a domain $\mathbb{D} = \{\mathscr{X}, \mathbb{P}^X\}$ has two components: a feature space $\mathscr{X}$ and a marginal probability distribution $\mathbb{P}^X$. Given $\mathbb{D}$, a task $\mathbb{T} = \{\mathscr{Y}, \mathbb{P}^{Y|X}\}$ has two components: the label space $\mathscr{Y}$ and the conditional probability distribution $\mathbb{P}^{Y|X}$. A common assumption behind most instance-based transfer learning approaches

is that the input instances of the source domain and the target domain have the same or very similar support, which means that the features for most instances have a similar range of values. Furthermore, the output labels of the source and target tasks are the same. This assumption ensures that knowledge can be transferred across domains via instances. According the definitions of a domain and a task, this assumption implies that, in instance-based transfer learning, the difference between domains/tasks is only caused by the differences of the marginal distribution of the features (i.e., $\mathbb{P}_s^X \neq \mathbb{P}_t^X$) or conditional probabilities (i.e., $\mathbb{P}_s^{Y|X} \neq \mathbb{P}_t^{Y|X}$).

When $\mathbb{P}_s^X \neq \mathbb{P}_t^X$ but $\mathbb{P}_s^{Y|X} = \mathbb{P}_t^{Y|X}$, we refer to the problem setting as noninductive transfer learning.[1] For example, suppose a hospital, either private or public, aims to learn a prediction model for a specific disease from its own patients' electronic medical records. Here we consider each hospital as a different domain. As the populations of patients of different hospitals are different, the marginal probabilities $\mathbb{P}^X$s are different across different domains. However, as the reasons that cause the specific disease are the same, the conditional probabilities $\mathbb{P}^{Y|X}$ across different domains remain the same. When $\mathbb{P}_s^{Y|X} \neq \mathbb{P}_t^{Y|X}$, we refer to the problem setting as inductive transfer learning. For instance, consider avian influenza virus as the specific disease in the previous example. As avian influenza virus has been evolving, the reasons causing avian influenza virus may change across different subtypes of avian influenza virus, for example, H1N1 versus H5N8. Here we consider learning a prediction model for each subtype of avian influenza virus for a specific hospital as a different task. As the reasons that cause different subtypes of avian influenza virus are different, the conditional probabilities $\mathbb{P}^{Y|X}$ are different across different tasks. In noninductive transfer learning, as the conditional probabilities across domains are the same, that is, $\mathbb{P}_s^{Y|X} = \mathbb{P}_t^{Y|X}$, it can be theoretically proven that, even without any labeled data in the target domain, an optimal predictive model can be learned from the source domain-labeled data and the target domain-unlabeled data. While in the inductive transfer learning case, as the conditional probabilities are different across tasks, a few labeled data in the target domain would then be required to exist to help transfer the conditional probability or the discriminative function from the source task to the target task. Since the assumptions of noninductive transfer learning and inductive transfer learning are different, the designs of instance-based transfer learning approaches for these two settings are different. In the following, we will review the motivations, basic ideas and representative methods for noninductive and inductive transfer learning in detail.

[1] Note that here we do not adopt the term "transductive transfer learning" used by Pan and Yang (2010) because the term "transductive" has been widely used to distinguish whether a model has an out-of-sample generalization ability, which may cause some confusion if used to define transfer learning problem settings.

2.2 Instance-Based Noninductive Transfer Learning

As mentioned earlier, in noninductive transfer learning, the source task and the target task are assumed to be the same, and the supports of the input instances across domains are assumed to be the same or very similar, that is, $\mathscr{X}_s = \mathscr{X}_t$. The only difference between domains is caused by the marginal distribution of input instances, that is, $\mathbb{P}_s^X \neq \mathbb{P}_t^X$. Under this setting, we are given a set of source domain-labeled data $\mathscr{D}_s = \{(\mathbf{x}_{s_i}, y_{s_i})\}_{i=1}^{n_s}$, and a set of target domain-unlabeled data $\mathscr{D}_t = \{(\mathbf{x}_{t_i})\}_{i=1}^{n_t}$. The goal is to learn a precise predictive model for the target domain unseen data.

In the following, we show that, under the assumptions in noninductive transfer learning, one is still able to learn an optimal predictive model for the target domain even without any target domain-labeled data. Suppose our goal is to learn a predictive model in terms of parameters θ_t for the target domain, based on the learning framework of empirical risk minimization (Vapnik, 1998), the optimal solution of θ_t can be learned by solving the following optimization problem.

$$\theta_t^* = \arg\min_{\theta_t \in \Theta} \mathbb{E}_{(\mathbf{x},y)\in\mathbb{P}_t^{X,Y}} [\ell(\mathbf{x}, y, \theta)], \tag{2.1}$$

where $\ell(\mathbf{x}, y, \theta)$ is a loss function in terms of the parameters θ_t. Since there are no target domain-labeled data, one cannot optimize (2.1) directly. It has been proven by Pan (2014) that, by using the Bayes' rule and the definition of expectation, the optimization (2.1) can be rewritten as follows,

$$\theta_t^* = \arg\min_{\theta_t \in \Theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_s^{X,Y}} \left[\frac{P_t(\mathbf{x}, y)}{P_s(\mathbf{x}, y)} \ell(\mathbf{x}, y, \theta_t) \right], \tag{2.2}$$

which aims to learn the optimal parameter θ_t^* by minimizing the weighted expected risk over source domain-labeled data. In noninductive transfer learning, as $\mathbb{P}_s^{Y|X} = \mathbb{P}_t^{Y|X}$, by decomposing the joint distribution $\mathbb{P}^{X,Y} = \mathbb{P}^{Y|X}\mathbb{P}^X$, we obtain $\frac{P_t(\mathbf{x},y)}{P_s(\mathbf{x},y)} = \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})}$. Hence, (2.2) can be further rewritten as

$$\theta_t^* = \arg\min_{\theta_t \in \Theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_s^{X,Y}} \left[\frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} \ell(\mathbf{x}, y, \theta_t) \right], \tag{2.3}$$

where a weight of a source domain instance $\mathbf{x}$ is defined as the ratio of marginal distributions of input instances between the target domain and the source domain at the data point $\mathbf{x}$. Given a set of source domain-labeled data $\{(\mathbf{x}_{s_i}, y_{s_i})\}_{i=1}^{n_s}$, by defining $\beta(\mathbf{x}) = \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})}$, an empirical approximation of (2.3) can be written as[2]

$$\theta_t^* = \arg\min_{\theta_t \in \Theta} \sum_{i=1}^{n_s} \beta(\mathbf{x}_{s_i}) \ell(\mathbf{x}_{s_i}, y_{s_i}, \theta_t), \tag{2.4}$$

Therefore, to properly reuse the source domain-labeled data to learn a target model, one needs to estimate the weight's $\{\beta(\mathbf{x}_{s_i})\}$. As shown in (2.4), to estimate $\{\beta(\mathbf{x}_{s_i})\}$,

[2] In practice, a regularization term is added to avoid model overfitting.

that is, density ratios, only input instances without labels from the source domain and the target domain are required. A simple solution to estimate $\{\beta(\mathbf{x}_{s_i})\}$ for each source domain instance is to first estimate $\mathbb{P}_t^X$ and $\mathbb{P}_s^X$, respectively, and then compute the ratio $\frac{P_t(\mathbf{x}_{s_i})}{P_s(\mathbf{x}_{s_i})}$ for each specific source domain instance $\mathbf{x}_{s_i}$. However, it is well known that density estimation itself is a difficult task (Tsuboi et al., 2009), especially when data are of high dimensions. In this way, the error caused by density estimation will be propagated to the density ratio estimation .

In the literature (Quionero-Candela et al., 2009), more promising solutions have been proposed to estimate $\frac{\mathbb{P}_t^X}{\mathbb{P}_s^X}$, directly bypassing the density estimation step. In the following sections, we introduce how to directly estimate the density ratio by reviewing several representative methods.

2.2.1 Discriminatively Distinguish Source and Target Data

One simple and effective approach to learn the weights is to transform the problem of estimating the marginal probability density ratio to the problem of distinguishing whether an instance is from the source domain or the target domain. This can be formulated as a binary classification problem with data instances from the source domain being labeled as 1 and those from the target domain being labeled as 0.

For example, Zadrozny (2004) proposes a rejection sampling-based method for correcting sample selection bias. The rejection sampling process is defined as follows. A binary random variable $\delta \in \{1,0\}$, which is called selection variable, is introduced. An instance $\mathbf{x}$ is sampled from the target marginal distribution $\mathbb{P}_t^X$ with probability $P_t(\mathbf{x})$, that is, $P_t(\mathbf{x}) = P(\mathbf{x}|\delta = 0)$. Similarly, $P_s(\mathbf{x})$ can be rewritten as $P_s(\mathbf{x}) = P(\mathbf{x}|\delta = 1)$. $\mathbf{x}$ is accepted by the source domain with probability $P(\delta = 1|\mathbf{x})$ or rejected with probability $P(\delta = 0|\mathbf{x})$. In mathematics, with the new variable δ, the density ratio for each data instance $\mathbf{x}$ can be formulated as

$$\frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} = \frac{P(\delta = 1)}{P(\delta = 0)} \frac{P(\delta = 0)}{P(\delta = 1)} \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})}, \tag{2.5}$$

where $P(\delta)$ is the prior probability of δ in the union data set of the source domain and the target domain. By using the Bayes, rule and the equivalent forms of $P_s(\mathbf{x})$ and $P_t(\mathbf{x})$ in terms of δ, (2.5) can be further reformulated as

$$\frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} = \frac{P(\delta = 1)}{P(\delta = 0)} \left(\frac{1}{P(\delta = 1|\mathbf{x})} - 1 \right).$$

Therefore, the density ratio for each source domain data instance can be estimated as $\frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} \propto \frac{1}{P_{s,t}(\delta=1|\mathbf{x})}$. To compute the probability $P(\delta = 1|\mathbf{x})$, we regard it as a binary classification problem and train a classifier to solve it. After calculating the ratio for each source data instance, a model can be trained by either reweighting each source data instance or performing importance sampling on the source data set.

Following the idea of Zadrozny (2004), Bickel et al. (2007) propose a framework