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 $\mathcal{X}$ and a marginal probability distribution $\mathbb{P}^X$, where each input instance $\mathbf{x} \in \mathcal{X}$. In general, if two domains are different, then they may have different feature spaces or different marginal probability distributions. Given a specific domain, $\mathbb{D} = \{\mathcal{X}, \mathbb{P}^X\}$, a *task* $\mathbb{T}$ consists of two components: a label space $\mathcal{Y}$ and a function $f(\cdot)$ (denoted by $\mathbb{T} = \{\mathcal{Y}, f(\cdot)\}$). 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, $\mathcal{Y} = \{-1, +1\}$, 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 $\mathcal{D}_s = \{(\mathbf{x}_{s_i}, y_{s_i})\}_{i=1}^{n_s}$ the *source domain labeled data*, where $\mathbf{x}_{s_i} \in \mathcal{X}_s$ is the data instance and $y_{s_i} \in \mathcal{Y}_s$ is the corresponding class label. Similarly, we denote by $\mathcal{D}_t = \{(\mathbf{x}_{t_i}, y_{t_i})\}_{i=1}^{n_t}$ the target domain labeled data, where the input $\mathbf{x}_{t_i}$ is in $\mathcal{X}_t$ and $y_{t_i} \in \mathcal{Y}_t$ is the corresponding

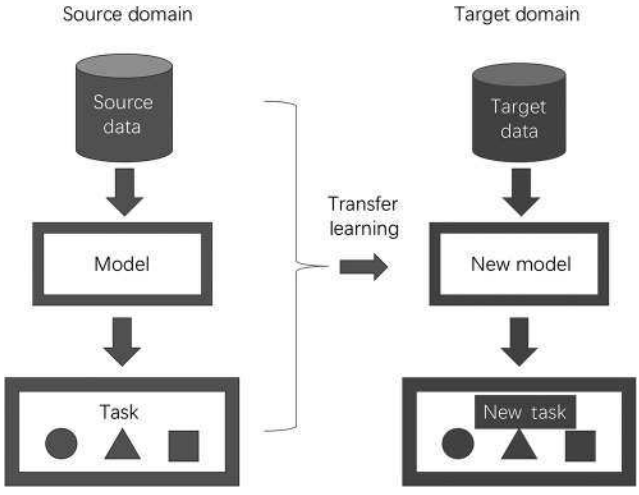

Figure 1.1 An illustration of a transfer learning process

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, $\mathbb{D} = \{\mathcal{X}, \mathbb{P}^X\}$, the condition $\mathbb{D}_s \neq \mathbb{D}_t$ implies that either $\mathcal{X}_s \neq \mathcal{X}_t$ or $\mathbb{P}^{X_s} \neq \mathbb{P}^{X_T}$. Similarly, as a task is defined as a pair of components $\mathbb{T} = \{\mathcal{Y}, \mathbb{P}^{Y|X}\}$, the condition $\mathbb{T}_s \neq \mathbb{T}_t$ implies that either $\mathcal{Y}_s \neq \mathcal{Y}_t$ 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 $\mathcal{X}_s \bigcap \mathcal{X}_t \neq \emptyset$ and $\mathcal{Y}_s = \mathcal{Y}_t$, 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 $\mathcal{X}_s \bigcap \mathcal{X}_t = \emptyset$ or $\mathcal{Y}_s \neq \mathcal{Y}_t$.

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