Domain Adaptation and Transfer Learning in Symbolic Regression

Tom Schmiedlechner

Information Engineering and Management
University of Applied Sciences Upper Austria

6th May, 2017

Contents

Abstract

As a result of the rise of big data applications and the need to make use of the collected data, the interest in machine learning applications has also grown considerably. Although this leads to more research on this topic and therefore positively affects both the speed and quality of the algorithms used, there is still high potential for improvements in both aspects, especially in the area of industrial applications with frequently changing data.

Despite not being the newest research topic, the interest in domain adaptation is currently increasing – among other things due to the latest developments in computing power. It is based on the idea of applying existing models to extended or modified application domains, instead of creating new ones for every use case. This topic is strongly related with the concept of transfer learning – two terms that are not always clearly delimited from each other. [PY10] [HXZ16]

To survey these terms, the content of this paper will focus on the description of domain adaptation, transfer learning and other related concepts on this field, like online and incremental learning. Aside from definitions and formalizations, the usage of adaptive learning methods in symbolic regression is the primary subject that will be discussed.

Chapter 1

Introduction

1.1 Motivation

The main purpose of data mining algorithms is to create models that describe patterns in given datasets as well as possible, and ultimately gain additional information from the data through this. As the generation of models is usually a complex task, it is highly recommended to include preexisting knowledge into this process. This can be achieved by the usage of domain adaptation in a historic context, specifically by using models from earlier runs in the modeling process. [GH88][CXZ15]

A field that is also strongly related to this historic approach is *incremental learning*. In classic machine learning, a model is trained with one (ideally big) initial learning dataset; in contrast to this, in incremental learning, the model is constantly altered according to the input data it handles. In most real-life scenarios, this is highly applicable, because data often arrives as a continuous stream. [GH16][Ben+10]

This is especially interesting if the processed data's attributes slowly change over time, a phenomenon called *concept drift*. If these changes are not considered in the modeling process, the accuracy of the output decreases over time – which is often hard to recognize, if the computed outputs are not constantly monitored.

Depending on the area of machine learning (regression, classification, etc.), the exact definition of a *model* varies. In general, it is generated from input data and describes this data in a way that allows to perform the desired operations on new datasets. In symbolic regression, a model is a mathematical formula, e.g. $e^{cos(x)} + 4x^2$. [Koz92]

As stated above, *domain adaptation* and *transfer learning* are often not clearly distinguished. In most definitions, domain adaptation describes the process of using already learned models for different, but somehow related domains – like using an e-mail spam filter created from a limited source group of persons to an extended or different target group. Too achieve this, a domain adaptation algorithm usually needs to adapt the learned model to work with the new data.

Transfer learning, on the other hand, is described as the usage of learned models for similar, but different data in the same application domain. An example would be the classification of products of all categories based on their reviews, with a model created only from a limited amount of product categories. [PC14]

Including preexisting models probably leads to different consequences for the modeling process and its results; on one side, the speed of the used algorithms could be increased, because several relevant aspects about the data are already known. On the other side, the model could reach a higher level of quality because of the additional information gained from the historic context. However, it is also conceivable that one of these measures – particularly the quality of the model – decreases, while the other one – e.g. the speed – profits from this method. [BBG16]

1.2 Objectives

In addition to general descriptions on the already briefly introduced concepts, the focus of this paper is primarily on the use of domain adaptation in symbolic regression.

The resulting research question will then be:

What are the consequences of using domain adaptation in symbolic regression algorithms, and to which possible use-cases can this combination be applied?

1.3 Methods and Limitations

The content of this paper will be a summary of the current state of the art of the topics described above. Therefore, the target methods are a literature research and merging the information provided by the different sources (starting with the references provided at the end of this document), under the aspect of the usage in symbolic regression problems.

To keep the focus of this paper on the relevant topics, it will be necessary to filter possible sources. Therefore, every book or paper cited will be evaluated against the following criteria:

- The main topic of the source is equal to one of the subjects of this paper.
- The topic is discussed in a general context, or in the context of symbolic regression.

Chapter 2

Domain adaptation and transfer learning

2.1 Overview

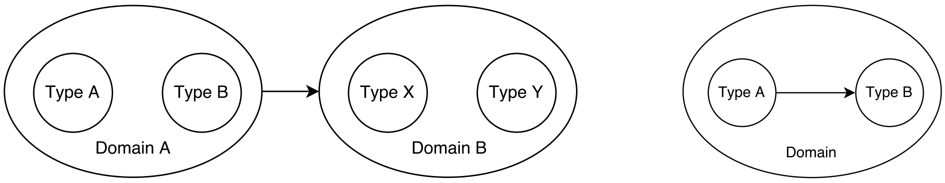
There are several definitions of domain adaptation and transfer learning in literature, therefore, the content of the following pages mainly focuses on finding an overall description of these two concepts and a method to clearly distinguish them. To achieve this, it is necessary to understand the formal characteristics of both ideas. Also, examples and practical applications will be used to clarify these definitions and distinctions.

2.1.1 Domain Adaptation

The essence of domain adaptation's most common definitions is that it describes the process of adapting models created from a source domain D_S to be used in a different, but somehow related target domain D_T. In supervised learning, this is particularly interesting because of the fact that a model is usually generated *before* or *while* the data to analyze is created. An example for this scenario could be the implementation of a spam filter, which is probably trained with pre-existing data from a group of users (= source domain), but only useful if it can also be applied to e-mails received by other users (=target domain). This implies that a data scientist (or model builder) usually only has access to a relatively small sample of the data that the model will be applied to in the future – simply because the data doesn't yet exist at the time of the model generation, or can't be accessed because it is private. [Ben+10]

The given example is also connected to the *concept drift* problem, which is often difficult to resolve or even detect. In long term applications of machine learning, the input data often changes over time – either slowly, or even abruptly. Especially the first case is hard to recognize if the output quality is not constantly monitored. When including a historic context in domain adaptation algorithms, these changes can be detected and handled properly – if the concept drift phenomenon is related to the specified domain. If this is not the case, incremental learning is another possibility (as described in section 2.3.2).

There are two possible approaches for the adaptation itself; the model can either be adapted to work in the scope of D_T by a specific domain adaptation algorithm,



(a) Domain adaptation **(b)** Transfer learning

Figure 2.1: Difference between domain adaptation and transfer learning.

or can be initially trained in a way that supports the target domain as well. [PC14]

2.1.2 Transfer learning

Transfer learning is commonly described as the usage of models for related, but different data in the same application domain (in the formalization introduced above, this means that the source domain D_S is equal to the target domain D_T). This can be especially interesting for processing large-scale datasets that contain different types of data – a concrete example is the classification of products of all categories based on their reviews, with a model created only from a limited amount of product categories. Since large web retailers like Amazon probably not only have to maintain a vast amount of product reviews, but also are adding new products and product categories continuously, the computational costs of creating new classification models without transfer learning would be very high. [PY10]

2.1.3 Differences and definitions

As already stated, the distinction of the two concepts is not trivial – also apparent from the example for domain adaptation described above. While it is very common in domain adaptation literature, it could also easily be classified as transfer learning by simply switching the definition of the *domain* from "user group" to "e-mails". This shows again how strongly connected these two concepts are.

Domain adaptation is also often treated as a sub-topic of transfer learning, as the common definition by *Pan et al.* states: [PY10]

Definition 1 (Transfer learning)**.** *Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$.*

If the condition $D_S \neq D_T$ is *true*, Definition 1 also applies to domain adaptation, which is treated as a part of transfer learning in this context. In the scope of this paper, *transfer learning* will therefore imply that $D_S \equiv D_T$ in Definition 1, and *domain adaptation* shall be defined as in Definition 2:

Definition 2 (Domain adaptation)**.** *Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$.*

In summary, this means that if the application domain changes during the adaptation process, we speak of domain adaptation – if it does not, usually transfer learning is the more appropriate term (see Figure 2.1). The diverse definitions are mainly based on the unclear definition of the *domain* term, hence it is essential to specify it in each discussion context. The clear definition is not only formally relevant, but also in terms of the used adaptation algorithms, which essentially differ between the two approaches. [PC14]

2.2 Applications

As mentioned above, domain adaptation and transfer learning are particularly interesting if the target domain (or the target data) is unknown at design time and/or expected to change over time. The given examples – a spam filter for domain adaptation and a product rating analysis for transfer learning – are most common in research, because they illustrate the topics very well, and are also suitable to demonstrate the differences between the two concepts. Nevertheless, there are several more *real-life* scenarios in which these concepts would fit very well, or even are already in use.

An interesting use case of domain adaptation that is currently investigated in is image recognition in changing visual domains. Since pictures are almost always taken under very different conditions, factors like post, angle and lighting may vary. *Hoffman et al.* presented a classification algorithm that applies domain adaptation principles to this field, and evaluated it with stock images from Amazon as source domain, and real world images as target domain[1]. [Hof+13]

Another field of application for domain adaptation is natural language processing, as shown by *Chan and Ng* in the context of *word sense disambiguation*. WSD is an open topic in natural language processing that describes the problem of detecting the meaning of a word in a specific context (usually a sentence). When changing the domain, e.g. from newspapers to scientific papers, the accuracy of a model trained for WSG operations usually drops because of the different usage of words in different domains. [CN07]

Pan and Yang also collected an extensive overview about most transfer learning and domain adaptation applications, including (but not limited to) the topics already mentioned above. Due to the shifting usage of the terms domain adaptation and transfer learning in literature, according to the definitions made in section 2.1.3, many examples labeled as *transfer learning* are actually more related to domain adaptation and vice versa in the context of this paper. [PY10]

[1] https://cs.stanford.edu/~jhoffman/domainadapt

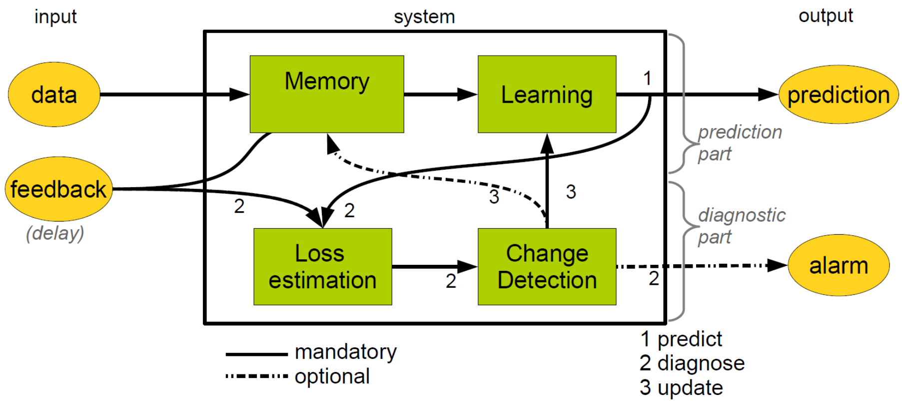


Figure 2.2: A generic schema for online adaptive learning algorithms. [Gam+14]

2.3 Related concepts

There are several connected concepts and terms to domain adaptation and transfer learning. Unfortunately, in most cases definitions tend to vary and overlap, as in the case of domain adaptation and transfer learning itself. This sections hence concentrates on the concepts of online and incremental learning and their delimitation of each other, since both topics are highly relevant in the context of concept drift and learning in long time periods. If the concept drift phenomenon affects the domain too, combinations of the described approaches could also become interesting.

2.3.1 Online learning

As described in section 2.1.1, concept drift can have a severe impact on the quality or speed of long-term machine learning applications. Online learning is an option to handle this problem of slowly or abruptly changing input data. While in classic machine learning, a model is trained with one (ideally big) initial learning dataset, in online learning, the model is constantly adjusted to the input data it handles. In most real-life scenarios, this is highly applicable, because data often arrives as a continuous stream (see figure 2.3.1).

The usual approach to achieve this is to specify some of the parameters of the model that will be altered during the online learning process. Depending on the algorithm used, these may vary, and therefore, this is the most crucial part in designing online learning algorithms (or adapting pre-existing ones). [Gam+14]

The disadvantage of online learning in the described form is that, due to the fact the all data has to be stored in the memory during the whole execution time, memory consumption tends to be very high. An option to bypass this problem is incremental learning, as described in section 2.3.2.

2.3.2 Incremental learning

Since online learning depends on the ability of constantly knowing all previous input data to generate model parameters, it is hardly usable in large datasets – a problem that has become more and more important due to the high popularity of big data applications and the subsequent collection of enormous amounts of data. Even with modern computational clusters (or *cloud computing*), it is hard and especially expensive to build environments with the required amount of RAM.

Incremental learning is an approach to circumvent these memory-based limitations of online learning of stream-based input data processing. To achieve this, instead of storing all previously received data fully detailed in the memory, incremental learning algorithms mostly rely on the previously generated models (and some additional meta data). Due to this, it is also possible to react relatively fast, even if the input data is abruptly changing. Working with streams in that way can be way more efficient than distributing the load over many machines, both in costs and in speed – mainly because it does not repeat the whole model generation process every time it should be adapted. [GH16]

Chapter 3

Domain adaptation in symbolic regression

3.1 Symbolic regression in machine learning

3.2 Advantages and disadvantages

3.3 Areas of application

Chapter 4

Conclusions

References

Literature

[BBG16] Adeleh Bitarafan, Mahdieh Soleymani Baghshah, and Marzieh Gheisari. “Incremental evolving domain adaptation”. *IEEE Transactions on Knowledge and Data Engineering* 28.8 (2016), pp. 2128–2141 (cit. on p. 2).

[Ben+10] Shai Ben-David et al. “A theory of learning from different domains”. *Machine learning* 79.1-2 (2010), pp. 151–175 (cit. on pp. 1, 3).

[CN07] Yee Seng Chan and Hwee Tou Ng. “Domain adaptation with active learning for word sense disambiguation”. In: *annual meeting-association for computational linguistics*. Vol. 45. 1. 2007, p. 49 (cit. on p. 5).

[CXZ15] Qi Chen, Bing Xue, and Mengjie Zhang. “Generalisation and domain adaptation in GP with gradient descent for symbolic regression”. In: *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE. 2015, pp. 1137–1144 (cit. on p. 1).

[Gam+14] João Gama et al. “A survey on concept drift adaptation”. *ACM Computing Surveys (CSUR)* 46.4 (2014), p. 44 (cit. on p. 6).

[GH16] Alexander Gepperth and Barbara Hammer. “Incremental learning algorithms and applications”. In: *European Symposium on Artificial Neural Networks (ESANN)*. 2016 (cit. on pp. 1, 7).

[GH88] David E Goldberg and John H Holland. “Genetic algorithms and machine learning”. *Machine learning* 3.2 (1988), pp. 95–99 (cit. on p. 1).

[Hof+13] Judy Hoffman et al. “Efficient learning of domain-invariant image representations”. *arXiv preprint arXiv:1301.3224* (2013) (cit. on p. 5).

[HXZ16] Edward Haslam, Bing Xue, and Mengjie Zhang. “Further investigation on genetic programming with transfer learning for symbolic regression”. In: *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE. 2016, pp. 3598–3605 (cit. on p. ii).

[Koz92] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*. Vol. 1. MIT press, 1992 (cit. on p. 1).

[PC14] Novi Patricia and Barbara Caputo. "Learning to learn, from transfer learning to domain adaptation: A unifying perspective". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1442–1449 (cit. on pp. 2, 4, 5).

[PY10] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359 (cit. on pp. ii, 4, 5).