

Title: Domain adaptation

URL: https://en.wikipedia.org/wiki/Domain_adaptation

PageID: 45390860

Categories: Category:Machine learning

Source: Wikipedia (CC BY-SA 4.0). Content may require attribution.

Domain adaptation is a field associated with machine learning and transfer learning . It addresses the challenge of training a model on one data distribution (the source domain) and applying it to a related but different data distribution (the target domain).

A common example is spam filtering , where a model trained on emails from one user (source domain) is adapted to handle emails for another user with significantly different patterns (target domain).

Domain adaptation techniques can also leverage unrelated data sources to improve learning. When multiple source distributions are involved, the problem extends to multi-source domain adaptation .

Domain adaptation is a specialized area within transfer learning. In domain adaptation, the source and target domains share the same feature space but differ in their data distributions. In contrast, transfer learning encompasses broader scenarios, including cases where the target domain's feature space differs from that of the source domain(s).

Classification of domain adaptation problems

Domain adaptation setups are classified in two different ways; according to the distribution shift between the domains, and according to the available data from the target domain.

Distribution shifts

Common distribution shifts are classified as follows:

Covariate Shift occurs when the input distributions of the source and destination change, but the relationship between inputs and labels remains unchanged. The above-mentioned spam filtering example typically falls in this category. Namely, the distributions (patterns) of emails may differ between the domains, but emails labeled as spam in the one domain should similarly be labeled in another.

Prior Shift (Label Shift) occurs when the label distribution differs between the source and target datasets, while the conditional distribution of features given labels remains the same. An example is a classifier of hair color in images from Italy (source domain) and Norway (target domain). The proportions of hair colors (labels) differ, but images within classes like blond and black-haired populations remain consistent across domains. A classifier for the Norway population can exploit this prior knowledge of class proportions to improve its estimates.

Concept Shift (Conditional Shift) refers to changes in the relationship between features and labels, even if the input distribution remains the same. For instance, in medical diagnosis, the same symptoms (inputs) may indicate entirely different diseases (labels) in different populations (domains).

Data available during training

Domain adaptation problems typically assume that some data from the target domain is available during training. Problems can be classified according to the type of this available data:

Unsupervised : Unlabeled data from the target domain is available, but no labeled data. In the above-mentioned example of spam filtering, this corresponds to the case where emails from the target domain (user) are available, but they are not labeled as spam. Domain adaptation methods can benefit from such unlabeled data, by comparing its distribution (patterns) with the labeled source domain data.

Semi-supervised : Most data that is available from the target domain is unlabelled, but some labeled data is also available. In the above-mentioned case of spam filter design, this corresponds to the case that the target user has labeled some emails as being spam or not.

Supervised : All data that is available from the target domain is labeled. In this case, domain adaptation reduces to refinement of the source domain predictor. In the above-mentioned example classification of hair-color from images, this could correspond to the refinement of a network already trained on a large dataset of labeled images from Italy, using newly available labeled images from Norway.

Formalization

Let X be the input space (or description space) and let Y be the output space (or label space). The objective of a machine learning algorithm is to learn a mathematical model (a hypothesis) $h : X \rightarrow Y$ able to attach a label from Y to an example from X . This model is learned from a learning sample $S = \{ (x_i, y_i) \in (X \times Y) \mid i = 1 \dots m \}$.

Usually in supervised learning (without domain adaptation), we suppose that the examples $(x_i, y_i) \in S$ are drawn i.i.d. from a distribution D_S of support $X \times Y$ (unknown and fixed). The objective is then to learn h (from S) such that it commits the least error possible for labelling new examples coming from the distribution D_S .

The main difference between supervised learning and domain adaptation is that in the latter situation we study two different (but related) distributions D_S and D_T on $X \times Y$ [citation needed]. The domain adaptation task then consists of the transfer of knowledge from the source domain D_S to the target one D_T . The goal is then to learn h (from labeled or unlabelled samples coming from the two domains) such that it commits as little error as possible on the target domain D_T [citation needed].

The major issue is the following: if a model is learned from a source domain, what is its capacity to correctly label data coming from the target domain?

Four algorithmic principles

Reweighting algorithms

The objective is to reweight the source labeled sample such that it "looks like" the target sample (in terms of the error measure considered).

Iterative algorithms

A method for adapting consists in iteratively "auto-labeling" the target examples. The principle is simple:

a model h is learned from the labeled examples;

h automatically labels some target examples;

a new model is learned from the new labeled examples.

Note that there exist other iterative approaches, but they usually need target labeled examples.

Search of a common representation space

The goal is to find or construct a common representation space for the two domains. The objective is to obtain a space in which the domains are close to each other while keeping good performances on the source labeling task.

This can be achieved through the use of Adversarial machine learning techniques where feature representations from samples in different domains are encouraged to be indistinguishable.

Hierarchical Bayesian Model

The goal is to construct a Bayesian hierarchical model $p(\mathbf{n})$, which is essentially a factorization model for counts \mathbf{n} , to derive domain-dependent latent representations allowing both domain-specific and globally shared latent factors.

Softwares

Several compilations of domain adaptation and transfer learning algorithms have been implemented over the past decades:

SKADA (Python)

ADAPT (Python)

TLlib (Python)

Domain-Adaptation-Toolbox (MATLAB)

References