THE UNIVERSITY OF SYDNEY

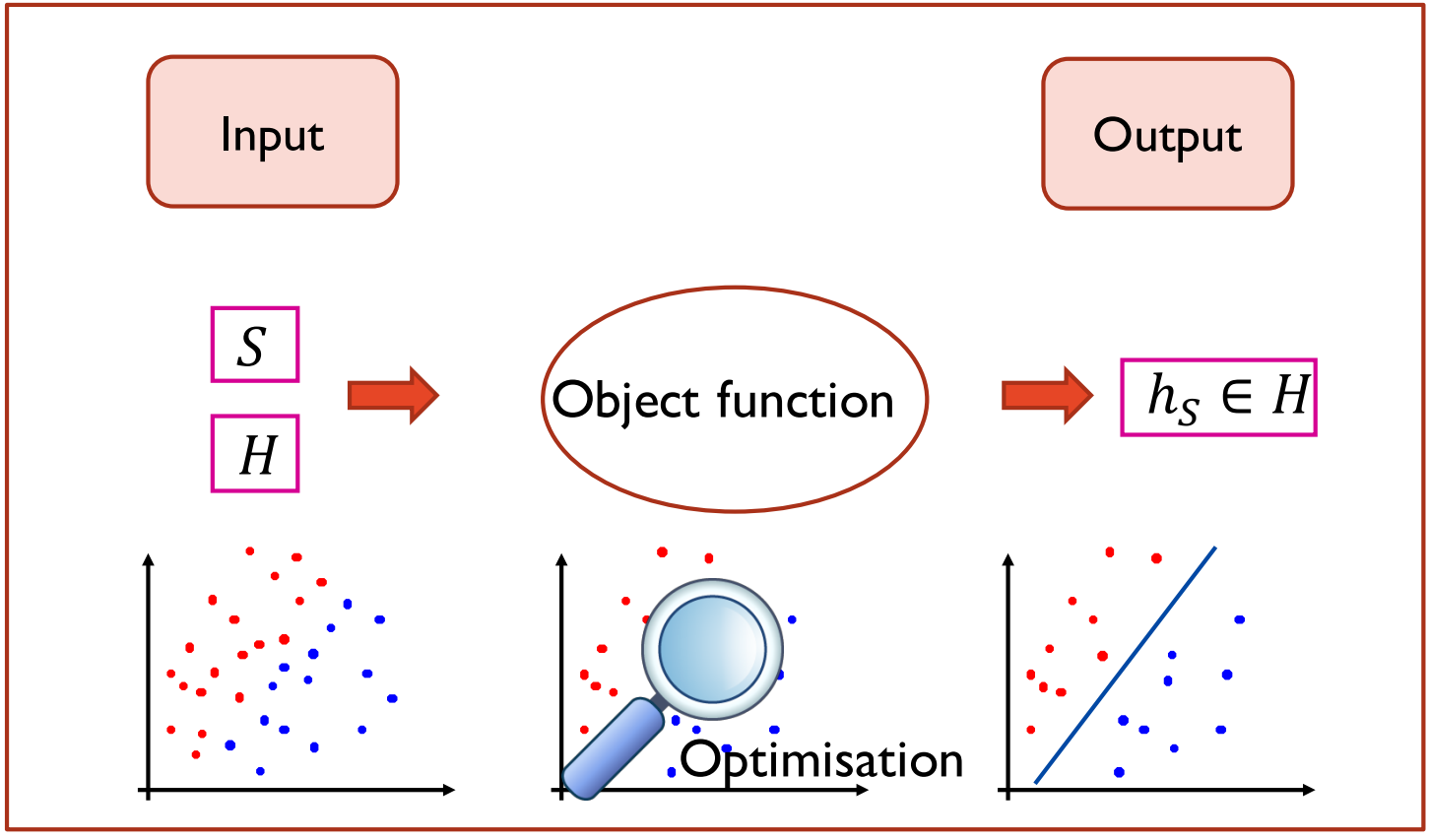
Advanced Machine Learning

(COMP 5328)

Domain Adaptation and Transfer Learning

Tongliang Liu

Machine Learning Flow Chart



Question 1

THE UNIVERSITY OF SYDNEY

- Why the hypothesis learned from the training dataset will have a good performance on the test dataset?

- Answer: We assume the training and test data have the same distribution. From the distribution perspective, they are the same. They should have the same error.

Question II

THE UNIVERSITY OF SYDNEY

- What is overfitting?

- Answer: The error on the training dataset is small while the error on the test dataset is large.

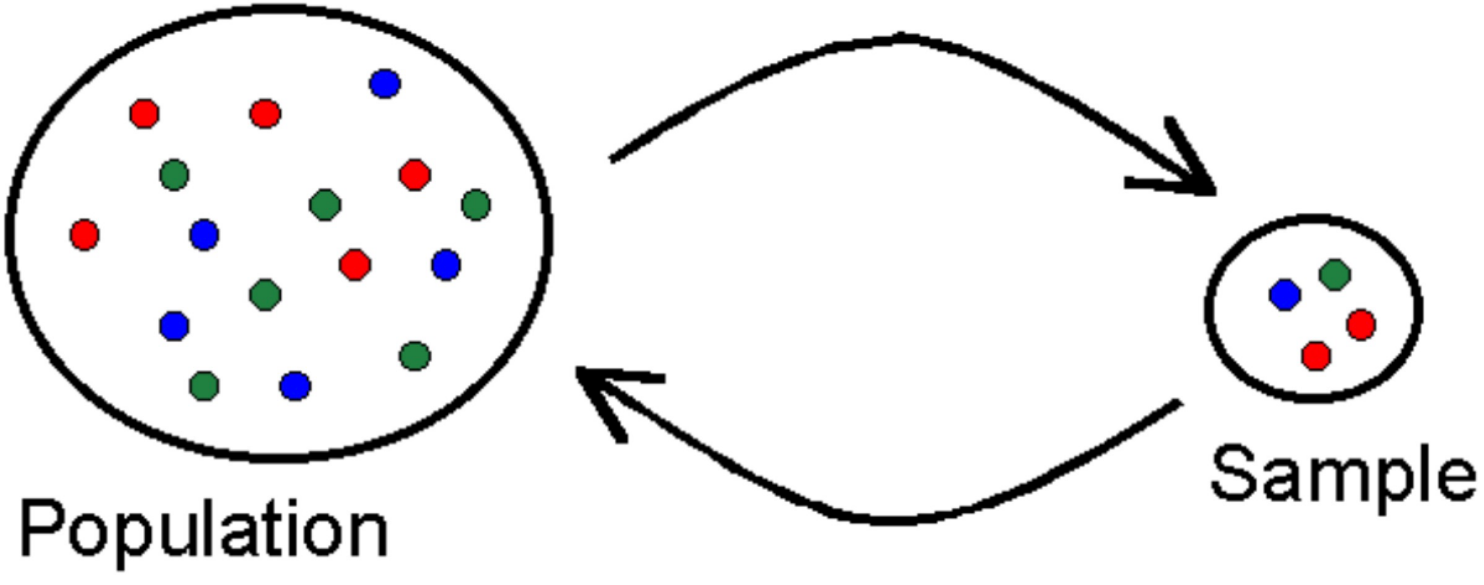
Question I vs Question II

- Do the answers to question I and question II conflict with each other? Why?

- Answer: No. They are about two different concepts: sample and population.

↑ training/test data

↑ distribution


Population
Sample

each task will correspond to a specific data distribution

Data distribution

THE UNIVERSITY OF SYDNEY

- If a task is given, we should exploit data information to design an effective learning algorithm.
- If we say two tasks are different because of data, it means that the population distributions of the data are different.
- If two tasks have the same population distribution of data, the training samples should be similar. In general, we should use the same learning algorithm for the tasks.

Notation

- Expected risk (expectation of the loss). $\mathbb{E}(A+B) = \mathbb{E}(A) + \mathbb{E}(B)$.

$$R(h) = \mathbb{E}[R_S(h)] = \mathbb{E}[\ell(X, Y, h)]$$
$$= \int_{(X,Y)} \ell(X, Y, h) p(X, Y) dX dY$$

- Empirical risk

each data, iid

$$R_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

where $S = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ is the training sample.

Notation

- The best function in the universal function space (target concept):

$$c = \arg\min_{h} R(h).$$

- The best function in the predefined hypothesis class:

$$h^* = \arg\min_{h \in H} R(h).$$

normally, $c \neq h^*$

if c is within H, $c = h^*$

- The hypothesis we can learn from data:

$$h_S = \arg\min_{h \in H} R_S(h) = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h).$$

Domain

THE UNIVERSITY OF SYDNEY

- Recall that a machine learning algorithm is a mapping to find a hypothesis to fit the data

$$\mathcal{A} : S \in (\mathcal{X} \times \mathcal{Y})^n \mapsto h_S \in H.$$

f(x) suppose domain
collection of all possible input value

- In machine learning, the distribution of the input data is called the domain.

Domain changes

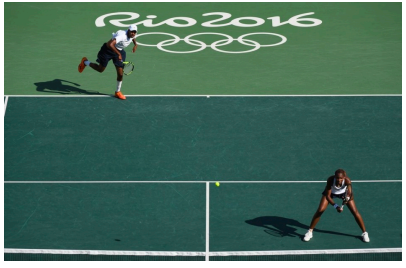
Art

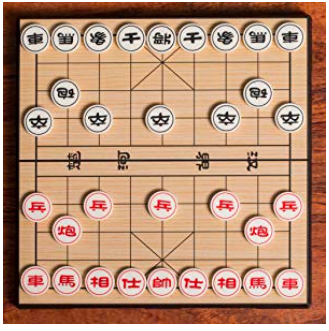
Surveillance

Aligned faces

“in the wild”

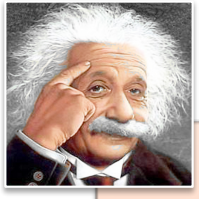
We human are good at finding the common knowledge and transferring the knowledge from one domain to another one.





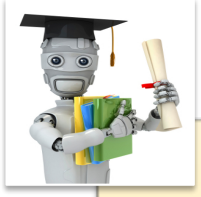
Can machines do in a similar way?

Human vs machine



Human

- Experience
- Choose a rule
- Make a decision



Machine

- A sample
- Choose a hypothesis
- Do a prediction

Domain adaptation and transfer learning

THE UNIVERSITY OF SYDNEY

Machine can also find the common knowledge between data and transfer the knowledge from one domain to another one. This relates two terms in machine learning: domain adaptation and transfer learning.

In machine learning, we can exploit training examples drawn from some related domain (the source domain) to improve the performance on the target domain.

Domain adaptation and transfer learning

THE UNIVERSITY OF SYDNEY

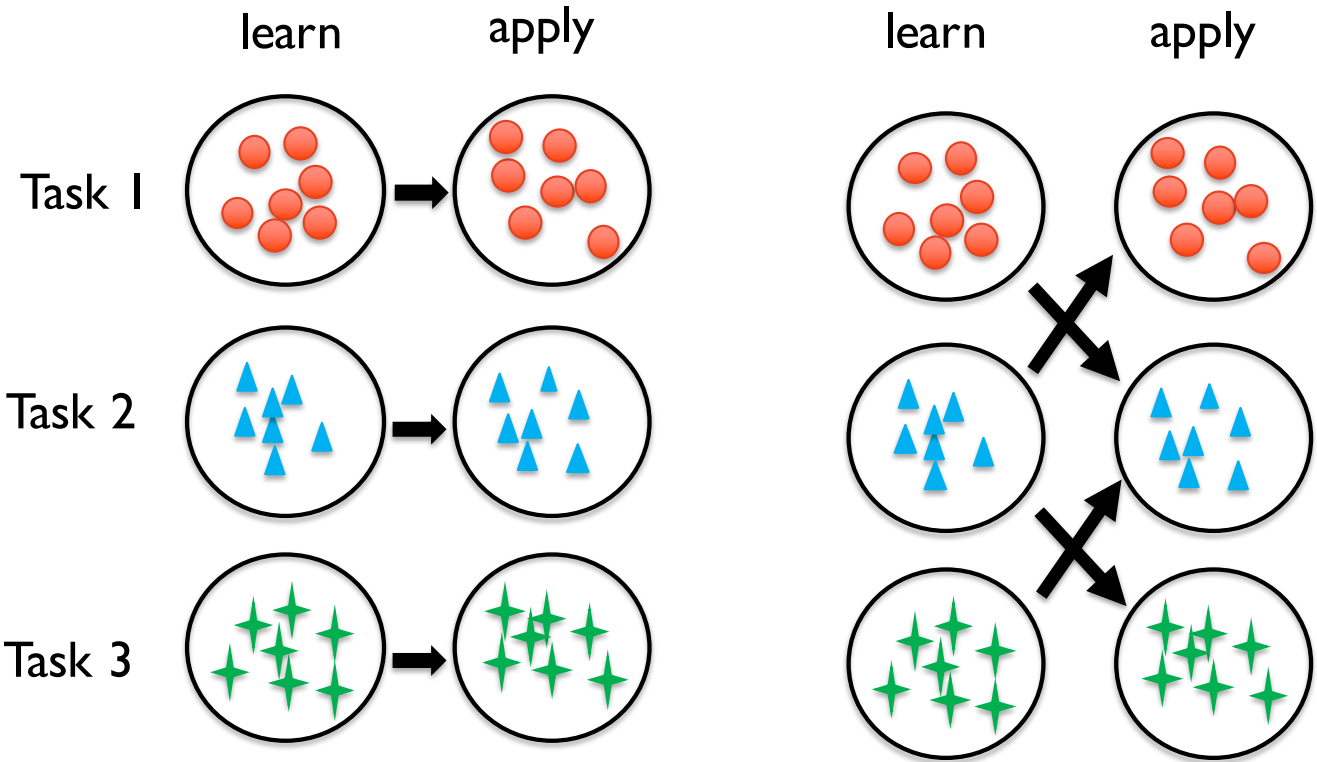
Domain adaptation: how to reduce the difference between the distributions of source and target domain data.

Transfer learning: extract knowledge from source domains and apply it to improve the learning performance in a target domain.

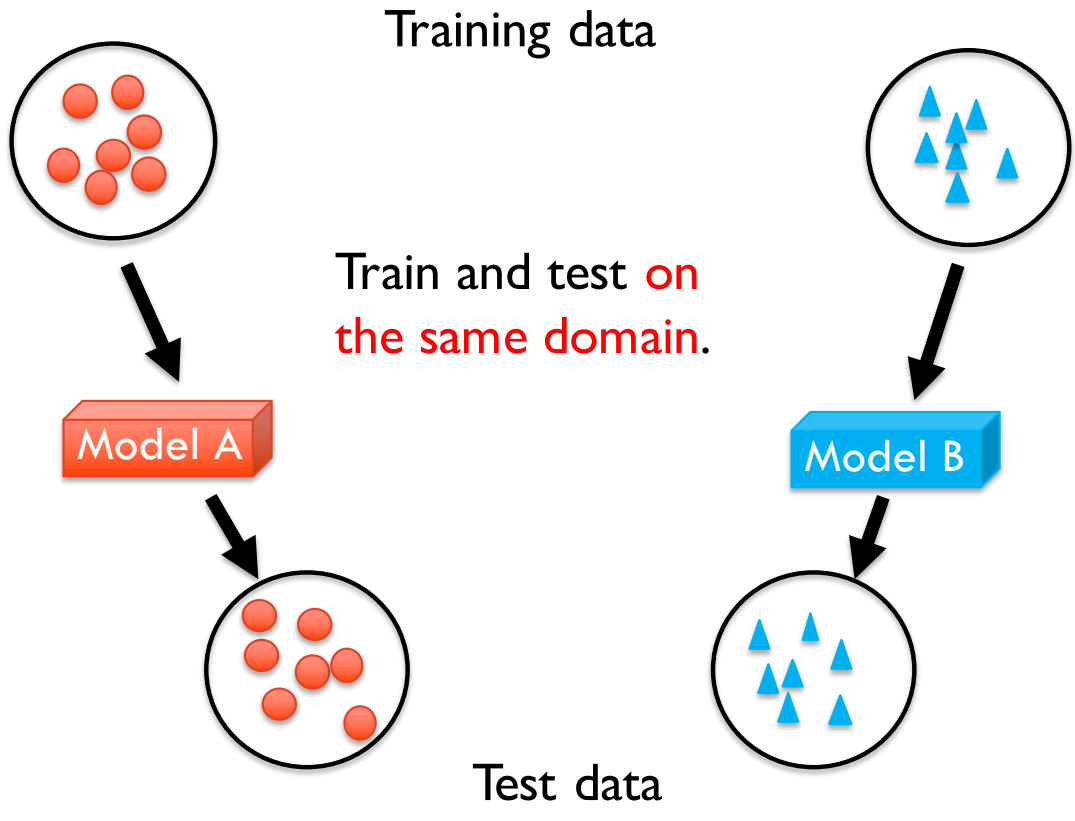
THE UNIVERSITY OF
SYDNEY

Transfer Learning

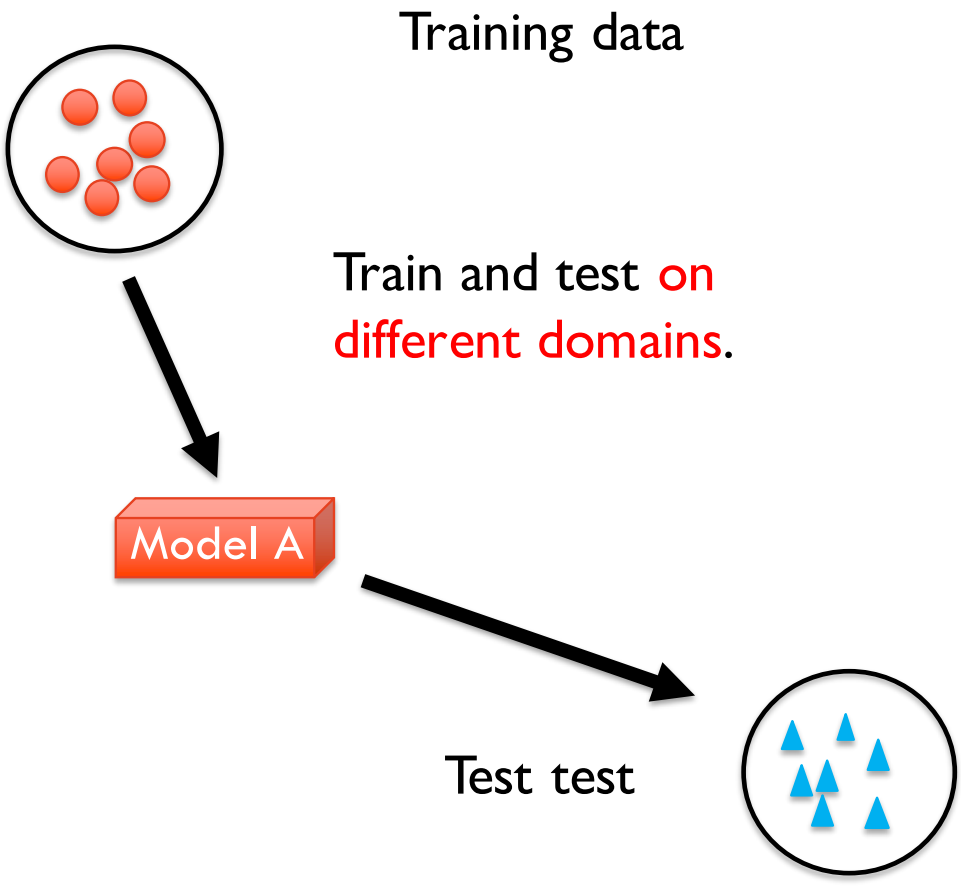
Human handles different tasks



Traditional machine learning

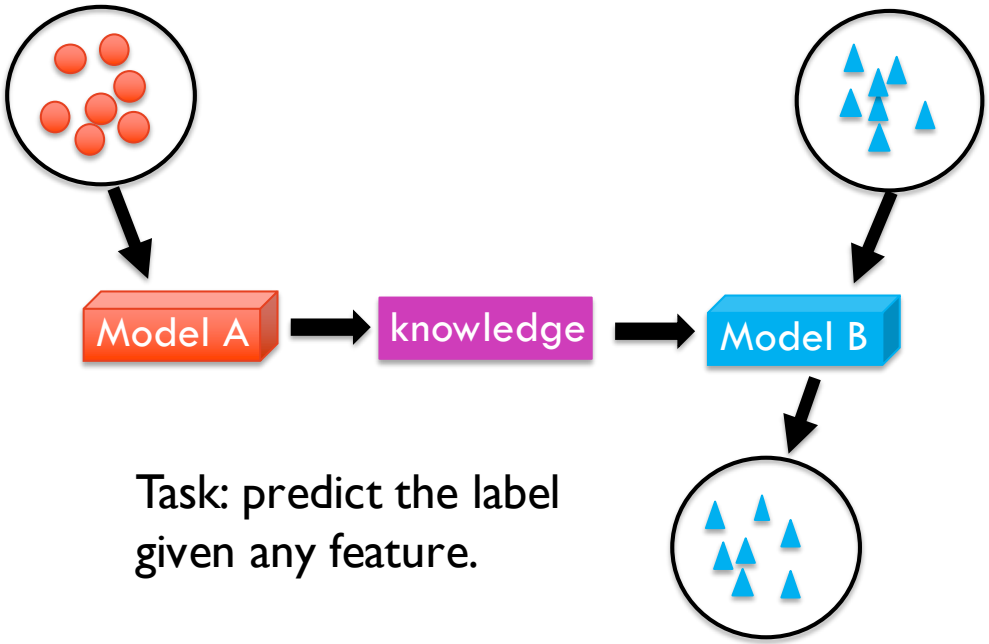


Transfer learning?



Let $\left\{\left(x_1^S, y_1^S\right), \ldots, \left(x_{n_S}^S, y_{n_S}^S\right)\right\}$ be the source domain data.

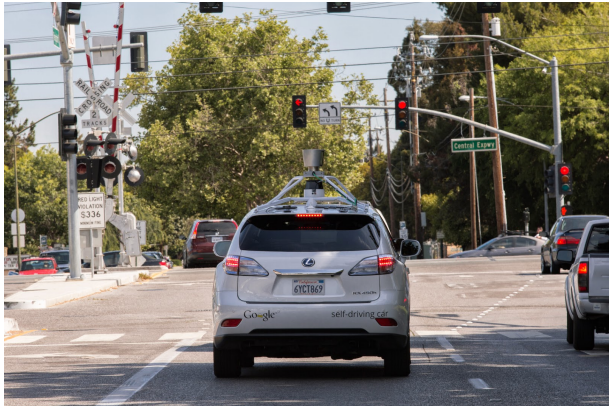
Let $\left\{\left(x_1^T, y_1^T\right), \ldots, \left(x_{n_T}^T, y_{n_T}^T\right)\right\}$ or $\left\{x_1^T, \ldots, x_{n_T}^T\right\}$ be the target domain data.



Task: predict the label given any feature.

Why do transfer learning?

In some domains, data, especially labeled data are expensive.



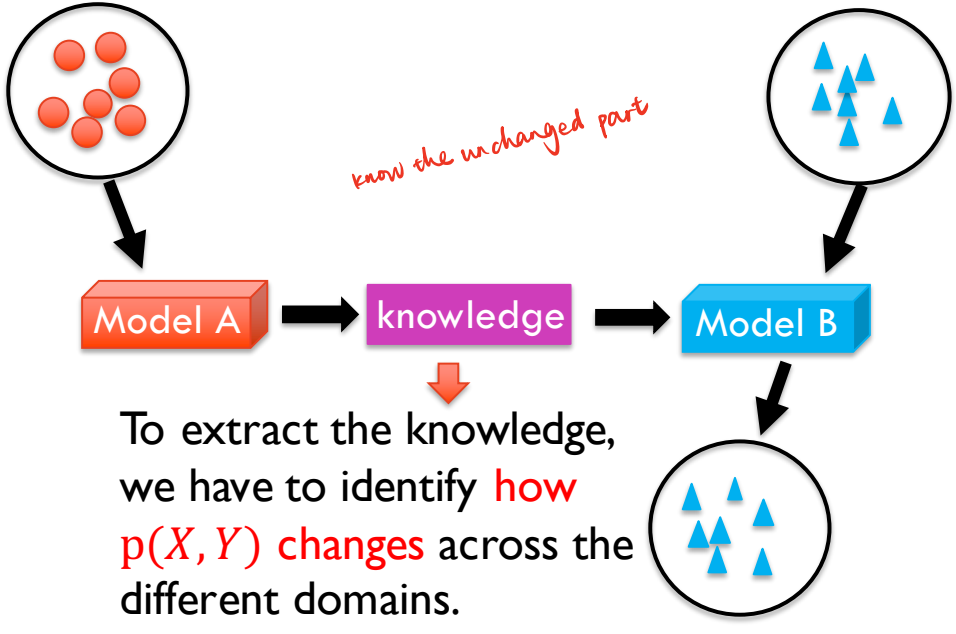
A Google self-driving car

Udacity's self-driving car simulator

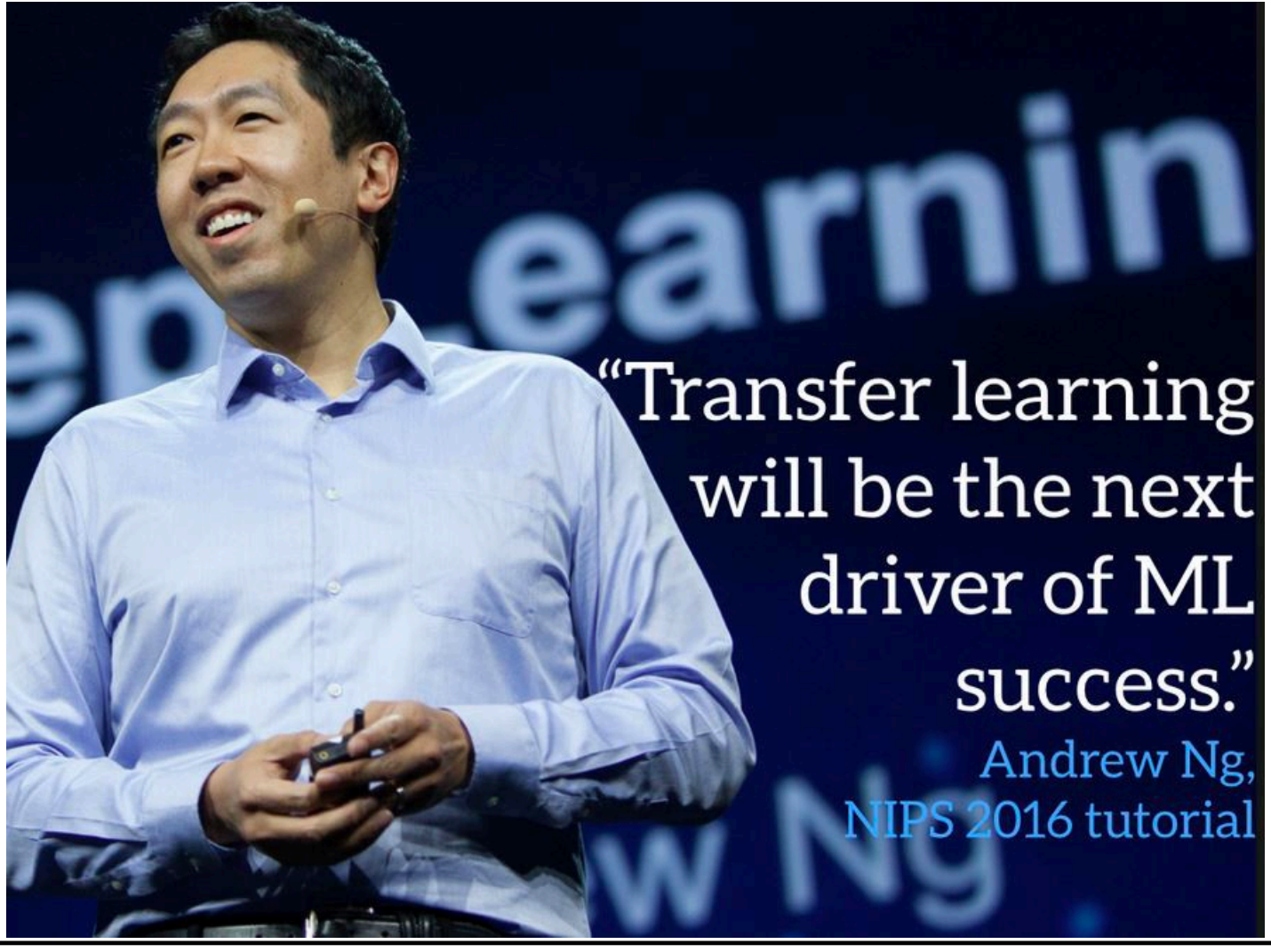
But in some related domains, lots of labelled data are available.

$\left\{\left(x_1^S, y_1^S\right), \ldots, \left(x_{n_S}^S, y_{n_S}^S\right)\right\}$ sampled from the source domain

$\left\{\left(x_1^T, y_1^T\right), \ldots, \left(x_{n_T}^T, y_{n_T}^T\right)\right\}$ sampled from the target domain. In most cases, y^T are not observed.



To extract the knowledge, we have to identify how $p(X, Y)$ changes across the different domains.


"Transfer learning will be the next driver of ML success."
Andrew Ng,
NIPS 2016 tutorial

Importance reweighting

Let denote $R^T(h) = \mathbb{E}_{(X,Y)\sim p_t(X,Y)}[\ell(X,Y,h)]$.

for target domain, data is limited, then the difference between empirical risk & expected risk is large

$\frac{1}{n}\sum_{i=1}^{n} \ell(x_i^{target}, y_i^{target}, h)$.

We have

$$
\begin{aligned}
R^T(h) &= \mathbb{E}_{(X,Y)\sim p_t(X,Y)}[\ell(X,Y,h)] \\
&= \int_{(X,Y)} \ell(X,Y,h) p_t(X,Y) dX dY \\
&= \int_{(X,Y)} \ell(X,Y,h) \frac{p_t(X,Y)}{p_s(X,Y)} p_s(X,Y) dX dY \\
&= \mathbb{E}_{(X,Y)\sim p_s(X,Y)}[\frac{p_t(X,Y)}{p_s(X,Y)} \ell(X,Y,h)] \\
&= \mathbb{E}_{(X,Y)\sim p_s(X,Y)}[\beta(X,Y)\ell(X,Y,h)],
\end{aligned}
$$

$n^S \to \infty$.

$\frac{1}{n}\sum \beta(x_i^S, y_i^S)\, \ell(x_i^S, y_i^S, h)$.

where the weights $\beta(X,Y) = p_t(X,Y)/p_s(X,Y)$ represent the changes across domains.

Importance reweighting

THE UNIVERSITY OF SYDNEY

If we know the value of $\beta(X, Y)$, we can use the source domain data to approximate the expected risk for the target domain.

$$
\begin{aligned}
R^T(h) &= \mathbb{E}_{(X,Y)\sim p_t(X,Y)}[\ell(X, Y, h)] \\
&= \int_{(X,Y)} \ell(X, Y, h) p_t(X, Y) dX dY \\
&= \int_{(X,Y)} \ell(X, Y, h) \frac{p_t(X, Y)}{p_s(X, Y)} p_s(X, Y) dX dY \\
&= \mathbb{E}_{(X,Y)\sim p_s(X,Y)}[\frac{p_t(X, Y)}{p_s(X, Y)} \ell(X, Y, h)] \\
&= \mathbb{E}_{(X,Y)\sim p_s(X,Y)}[\beta(X, Y)\ell(X, Y, h)]
\end{aligned}
$$

Importance reweighting

THE UNIVERSITY OF SYDNEY

If we know the value of $\beta(X, Y)$, we can use the source domain data to approximate the expected risk for the target domain.

Let $\{(x_1^S, y_1^S), \ldots, (x_{n_S}^S, y_{n_S}^S)\}$ be the training sample of source domain. We can approximate $R^T(h)$ by

$$\frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_i^S, y_i^S) \ell(x_i^S, y_i^S, h).$$

We therefore can learn the hypothesis for the target domain by exploiting examples from the source domain.

THE UNIVERSITY OF
SYDNEY

Domain Adaptation

Domain Adaptation

THE UNIVERSITY OF SYDNEY

In machine learning, a specific domain can be regarded as a specific joint distribution $p(X, Y)$.

- If $Y \in \{1, 2, \ldots, C\}$ the problem is called classification
- If $Y \in \mathbb{R}$ the problem is called regression

Domain Adaptation

THE UNIVERSITY OF SYDNEY

For the source domain, we have a data distribution $p_s(X, Y)$;
For the target domain, we have a data distribution $p_t(X, Y)$.

Under what conditions can we directly adapt a classifier trained on the source domain for use in the target domain?

$p_s(X, Y) = p_t(X, Y)$?

Domain Adaptation

THE UNIVERSITY OF SYDNEY

If the source and target domains are different, we should reduce the difference between the distributions while keeping as much information as possible.

Kernel mean matching

THE UNIVERSITY OF SYDNEY

Denote $\phi : X \to \mathcal{H}$, where $\mathcal{H}$ is a RKHS (Reproducing Kernel Hilbert Space) with the kernel function $K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$ and $\langle \cdot, \cdot \rangle$ is the inner product operator.

eg $\langle a, b \rangle = a^T b$

Let $\mu(p(X)) = \mathbb{E}_{X \sim p(X)}[\phi(X)]$, (mapping)

where $p(X)$ is a marginal distribution on the feature space.

if $\mu(p_1) = \mu(p_2)$ $\Rightarrow p_1 = p_2$

one-to-one

$\mu_1 \Rightarrow p_1$, p_2 ⋮ ; $p_1 \to \mu_m$, $\to \mu_n$ ⋮

↑ different μ correspond to different p eg. Gaussian kernel

The expectation μ is a bijective function if K is a universal kernel.

Theorem 1.2 in

Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., & Schölkopf, B. (2009). Covariate shift by kernel mean matching. http://www.gatsby.ucl.ac.uk/~gretton/papers/covariateShiftChapter.pdf

Kernel mean matching

define $p(x) = \frac{p_t(x)}{p_s(x)}$ by def $\mu(p(x)) = E_{x \sim p(x)}[\phi(x)]$

$\downarrow = \mu[p(x)\phi_s(x)]$

If we have

$$\mu(p_t(X)) = \mathbb{E}_{X \sim p_s(X)}[\beta(X)\phi(X)]$$

$= \int p(x)\phi(x)p_s(x)dx$

$= \int p_t(x)\phi(x)dx$

and $\beta(X) \geq 0$, $\mathbb{E}_{X \sim p_s(X)}[\beta(X)] = 1$,

what is the relationship between $\beta(X)p_s(X)$ and $p_t(X)$?

They are equal to each other. Because we have

$$\mu(\beta(X)p_s(X)) = \mu(p_t(X)).$$

Kernel mean matching

THE UNIVERSITY OF SYDNEY

We can learn the weights

$$\min_{\beta} \|\mu(p_t(X)) - \mathbb{E}_{X\sim p_s(X)}[\beta(X)\phi(X)]\|^2$$
$$\text{subject to } \beta(X) \geq 0, \mathbb{E}_{X\sim p_s(X)}[\beta(X)] = 1.$$

If we only have training samples from different domains, e.g., $\{x_1^S, \ldots, x_{n_S}^S\} \sim p_s(X)^{n_S}$ and $\{x_1^T, \ldots, x_{n_T}^T\} \sim p_t(X)^{n_T}$, how can we learn $\beta(X)$?

Kernel mean matching

THE UNIVERSITY OF SYDNEY

Since we cannot calculate the expectation, we will use the empirical mean to approximate them, e.g.,

$$\min_{\beta} \left\| \frac{1}{n_T} \sum_{i=1}^{n_T} \phi(x_i^T) - \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_i^S)\phi(x_i^S) \right\|^2$$

$$\text{subject to } \beta(x_i^S) \geq 0, \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_i^S) = 1.$$

THE UNIVERSITY OF SYDNEY

Two transfer learning models

Transfer learning models

THE UNIVERSITY OF SYDNEY

Let $\{(x_1^S, y_1^S), \ldots, (x_{n_S}^S, y_{n_S}^S)\}$ be the training sample of source domain. We can approximate $R^T(h)$ by

$$\frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_i^S, y_i^S) \ell(x_i^S, y_i^S, h).$$

Our target is to learn $\beta(X, Y)$.

We introduce two models, where $\beta(X, Y)$ can be effectively learned.

The product rule of probability

$$p(X, Y) = p(Y|X)p(X) = p(X|Y)p(Y)$$

Covariate shift model

THE UNIVERSITY OF SYDNEY

labelling information

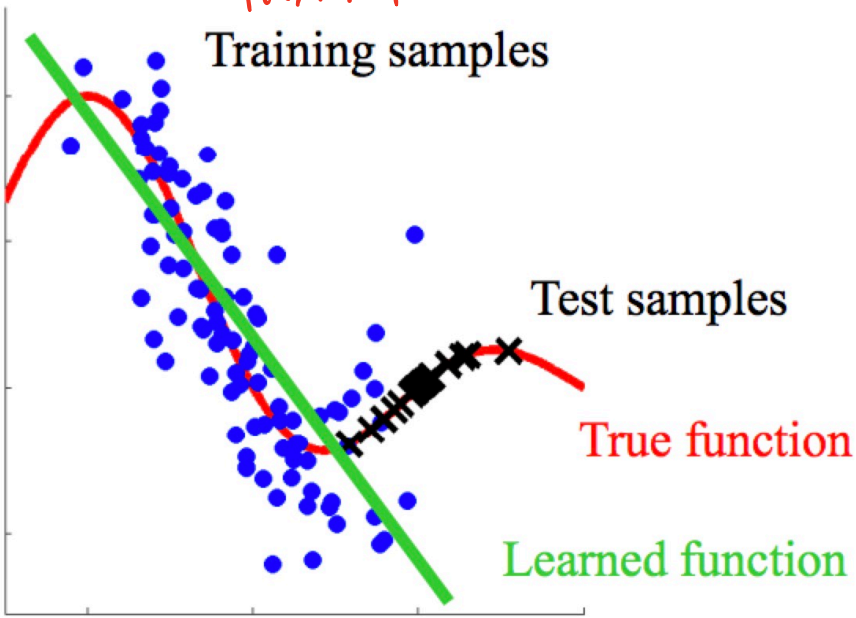
In this model, we assume that $p_t(Y|X) = p_s(Y|X)$ and that

$p_s(X) \neq p_t(X)$.

$p_s(X,Y) = p_s(Y|X) \cdot p_s(X)$

$p_t(X,Y) = p_t(Y|X) \cdot p_t(X)$

covariate shift setting
labeling information is the same
marginal distribution is different



http://iwann.ugr.es/2011/pdf/InvitedTalk-FHerrera-IWANN11.pdf

Covariate shift model

THE UNIVERSITY OF SYDNEY

If we assume that $p_t(Y|X) = p_s(Y|X)$, we have

$$\beta(X,Y) = \frac{p_t(X,Y)}{p_s(X,Y)} = \frac{p_t(Y|X)p_t(X)}{p_s(Y|X)p_s(X)} = \frac{p_t(X)}{p_s(X)} = \beta(X).$$

Note that $\beta(X)$ can be learned by kernel mean matching.

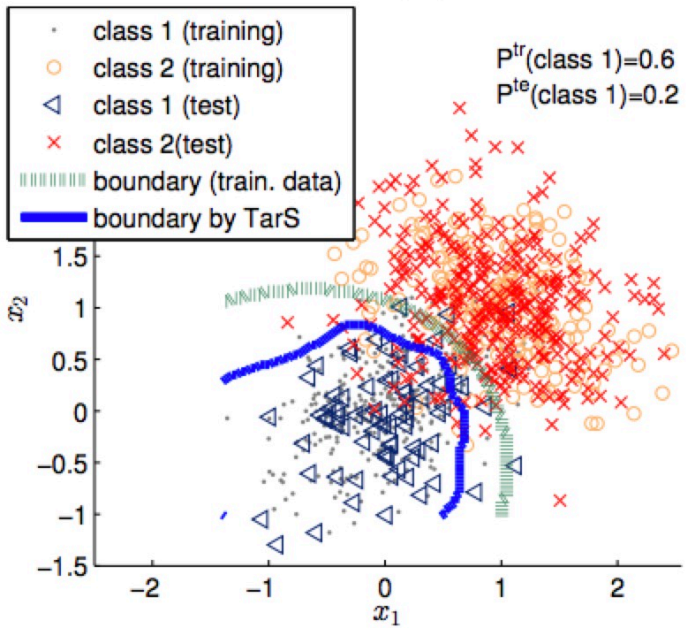
Target shift model

$p_s(X,Y) = p_s(X|Y) \cdot p_s(Y)$

$p_t(X,Y) = p_t(X|Y) \cdot p_t(Y)$

In this model, we assume that $p_t(X|Y) = p_s(X|Y)$ and that $p_t(Y) \neq p_s(Y)$.

distributions of Y are different in training and testing



Zhang, K., Schölkopf, B., Muandet, K., & Wang, Z. (2013, February). Domain adaptation under target and conditional shift. In International Conference on Machine Learning (pp. 819-827).

Target shift model

THE UNIVERSITY OF SYDNEY

If we further assume that $p_t(X|Y) = p_s(X|Y)$, we have

$$\beta(X,Y) = \frac{p_t(X,Y)}{p_s(X,Y)} = \frac{p_t(X|Y)p_t(Y)}{p_s(X|Y)p_s(Y)} = \frac{p_t(Y)}{p_s(Y)} = \beta(Y).$$

Note that $\beta(Y)$ is not easy to learn if the target domain does not have any labels. How to deal with this problem?

try both of them

train /val to train the model

Target shift model

We will use kernel mean matching to learn $\beta(Y)$.

We have

$$p_t(Y) = \beta(Y)p_s(Y)$$

from previous $\beta(Y) = \frac{p_t(Y)}{p_s(Y)}$.

$$p_t(X) = \int p_t(X|Y)\beta(Y)p_s(Y)dY = \int p_s(X|Y)\beta(Y)p_s(Y)dY$$

$= \int p_t(X|Y) \cdot p_t(Y)\, dY$

$p_s(X|Y)$: by condition

We can estimate $\beta(Y)$ by matching the distributions $p_t(X)$ and $\int p_s(X|Y)\beta(Y)p_s(Y)dY$.

Target shift model

THE UNIVERSITY OF SYDNEY

We have the following model

$= \mathbb{E}_{x \sim p_t(x)}[\phi(x)]$ problem: don't know $\mathbb{E}$

$$\min_{\beta} \|\mu(p_t(X)) - \mathbb{E}_{Y \sim p_s(Y)}[\mu(p_s(X|Y))\beta(Y)]\|^2$$

$$\text{subject to } \beta(Y) \geq 0, \mathbb{E}_{Y \sim p_s(Y)}[\beta(Y)] = 1.$$

$\mu(p(x)) = \mathbb{E}_{x \sim p(x)}[\phi(x)]$

Empirically, weighted sum

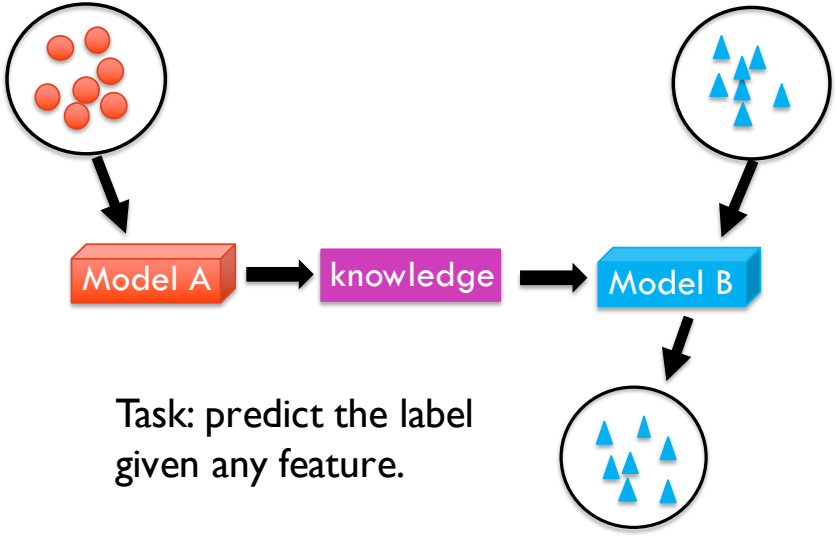
$$\min_{\beta} \left\| \frac{1}{n_T} \sum_{i=1}^{n_T} \phi(x_i^T) - \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(y_i^S) \hat{\mu}(p_s(X|y_i^S)) \right\|^2$$

$$\text{subject to } \beta(y_i^S) \geq 0, \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(y_i^S) = 1.$$

Real-world problems

Let $\left\{\left(x_1^S, y_1^S\right), \ldots, \left(x_{n_S}^S, y_{n_S}^S\right)\right\}$ be the source domain data.

Let $\left\{x_1^T, \ldots, x_{n_T}^T\right\}$ be the target domain data.

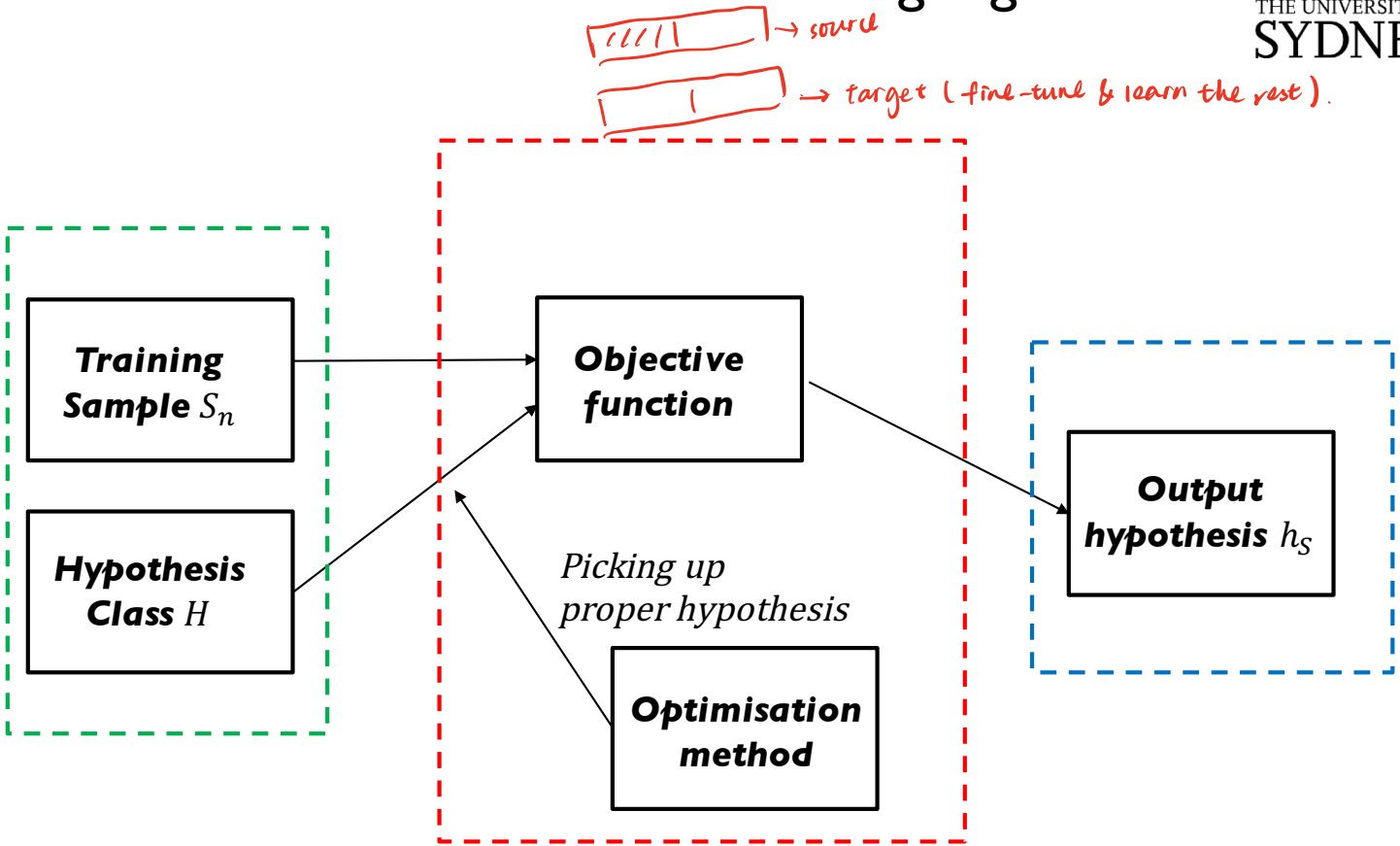


Which model can be employed to solve the problem? Covariate shift model or target shift model?

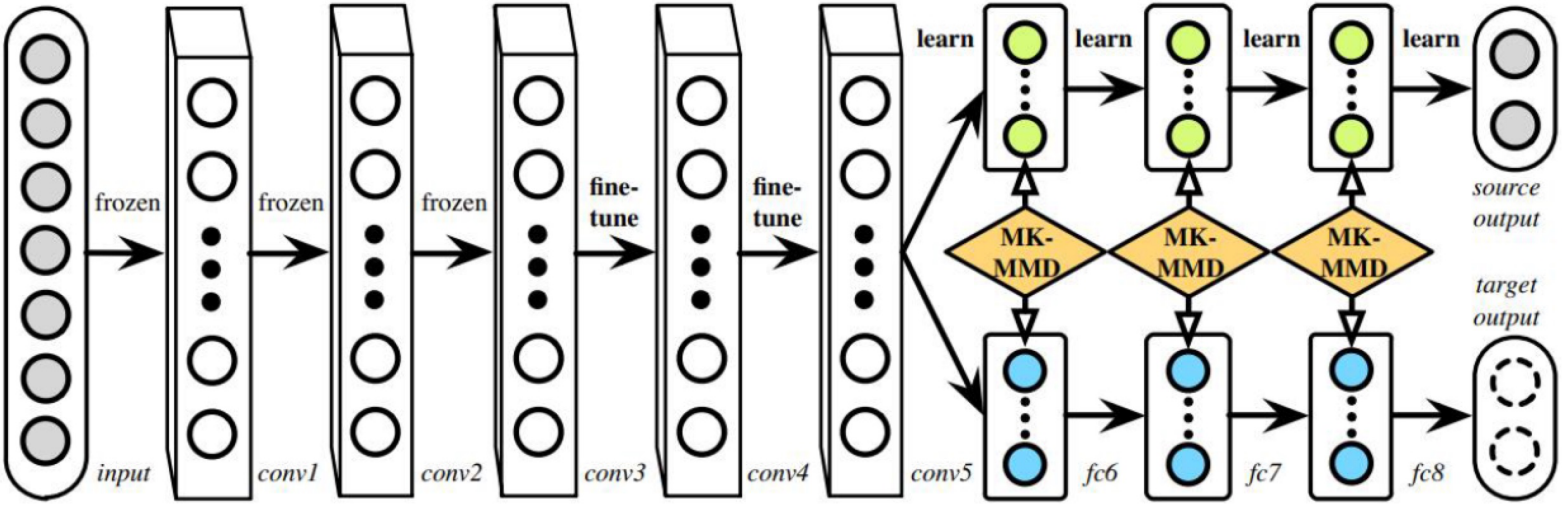
THE UNIVERSITY OF SYDNEY

Research topic (State-of-the art methods)

Some Elements of Machine Learning Algorithms

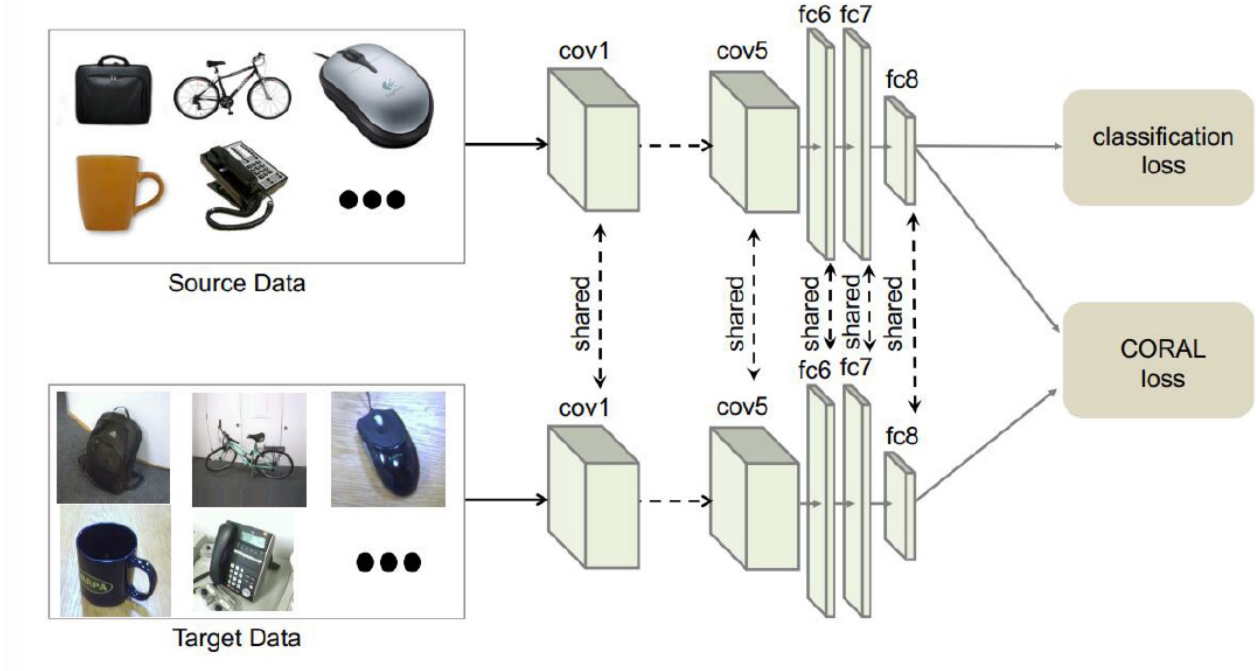


Deep Adaptation Networks



Long, M., Cao, Y., Wang, J., & Jordan, M. I. (2015). Learning transferable features with deep adaptation networks. arXiv preprint arXiv:1502.02791.

Deep CORrelation ALignment



Sun, B., & Saenko, K. (2016, October). Deep coral: Correlation alignment for deep domain adaptation. In European Conference on Computer Vision (pp. 443-450). Springer, Cham.