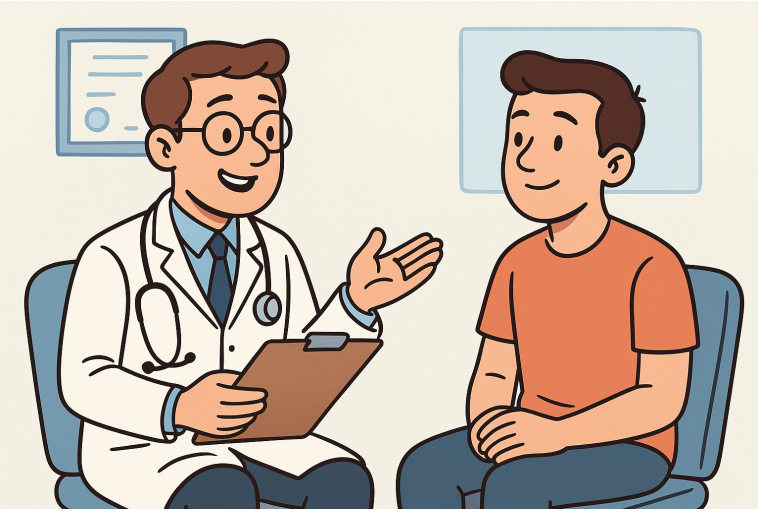
Importance Weighting for Unsupervised Domain Adaptation

A. de Mathelin, M. Atiq

Sloan Kettering Institute - CEA

ECAS 2025

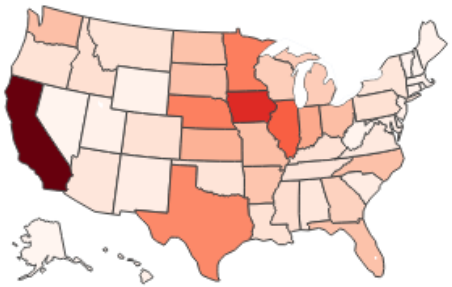
Importance Weighting Use Cases



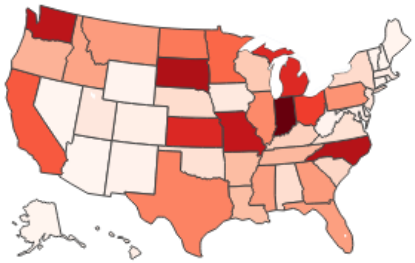
Medical Applications : Some groups of patients are under represented

Rare event probability estimation

Surveyed



Full



Polls :The distribution of the surveyed population differs from the full population

Definition 1 (Unsupervised Domain Adaptation (UDA))

Let's consider a feature space $\mathcal{X}$ and a label space $\mathcal{Y}$. We define $P_S(X, Y)$ and $P_T(X, Y)$ the respective source and target joint distributions over $\mathcal{X} \times \mathcal{Y}$.

We call **Unsupervised Domain Adaption** the learning setting composed of :

- A source **labeled** set $\mathcal{S} = \{(x_1, y_1), ..., (x_m, y_m)\} \in \mathcal{X} \times \mathcal{Y}$ where $x_i \sim P_S(X)$ and $y_i \sim P_S(Y|X = x_i)$
- A target **unlabeled** set $\mathcal{T}_\mathcal{X} = \{x'_1, ..., x'_m\} \in \mathcal{X}$ where $x'_j \sim P_T(X)$

Example

Patient ID	Patient variable (X)	Clinical variable (Y)
0	0.764052	0.427822
1	-0.599843	0.420066
2	-0.021262	-0.385311
...	...	...
27	-1.187184	1.097221
28	0.532779	-0.002597
29	0.469359	0.996585

30 rows × 2 columns

Figure – We consider a simple 1D regression task, where the learner is interested in the correlation between a specific patient variable X (eg : age, weight, gene expression...) and a clinical variable Y (eg : disease evolution...). The learner has collected 30 samples and want to fit a linear model $Y = aX + b$.

Example

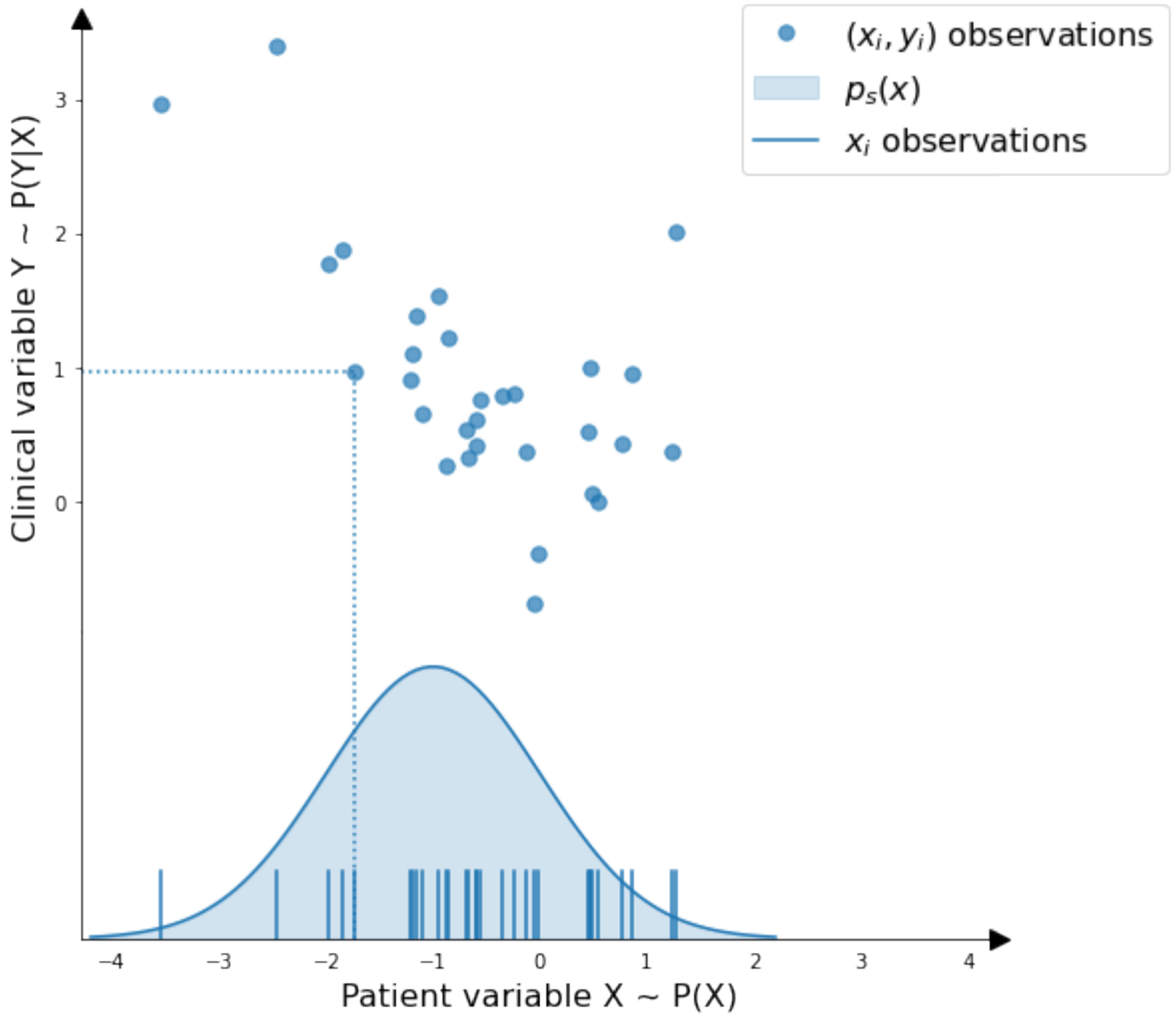


Figure – In practice, the learner assume that the collected sample is independently and identically distributed from an unknown source input distribution $P_S(X)$.

Example

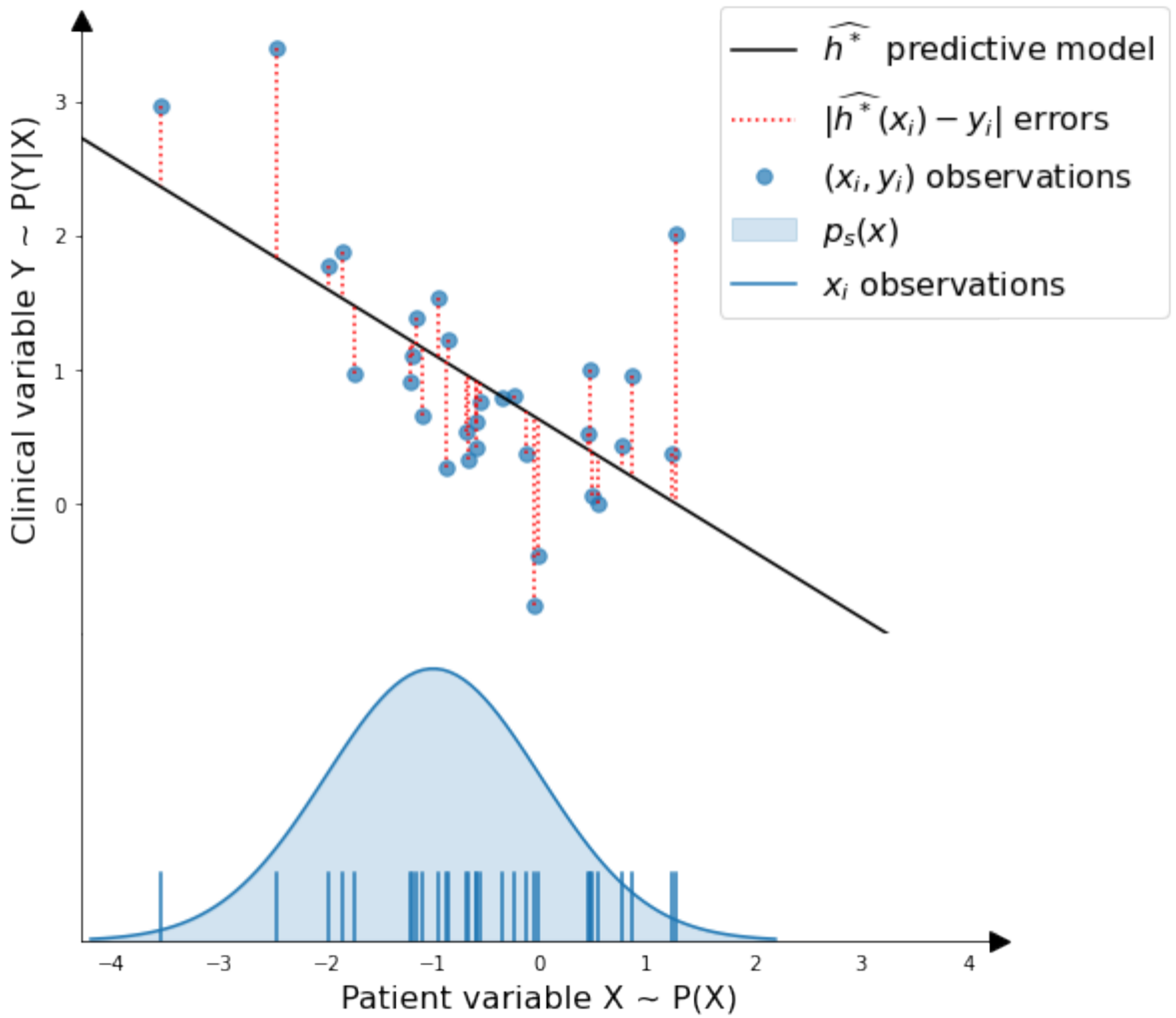


Figure – Based on the (x_i, y_i) observations, the learner can fit a linear model to estimate the correlation between the two variables.

Example

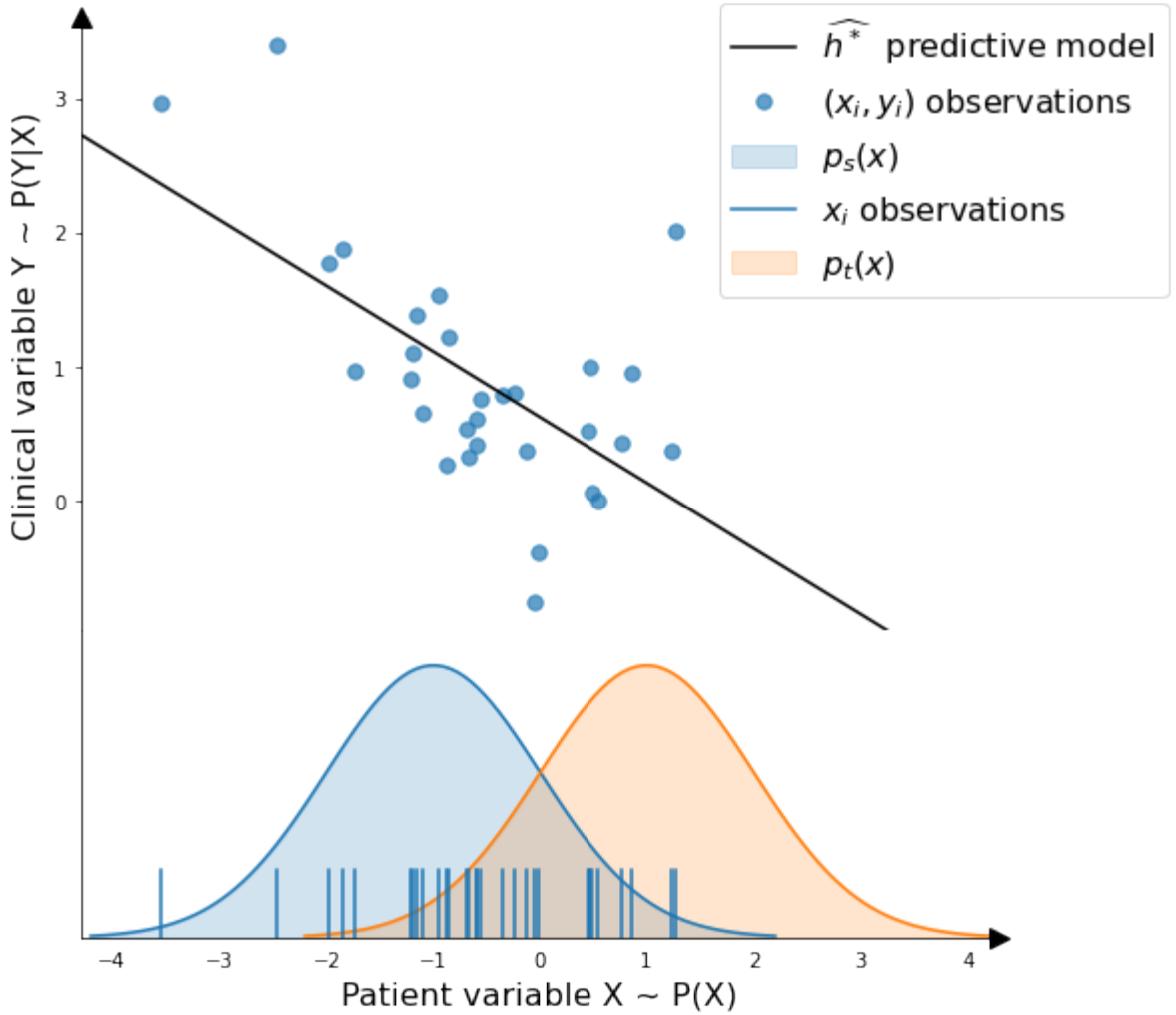


Figure – In the transfer learning scenario, the source collected sample is not representative of the targeted population (here in orange)

Example

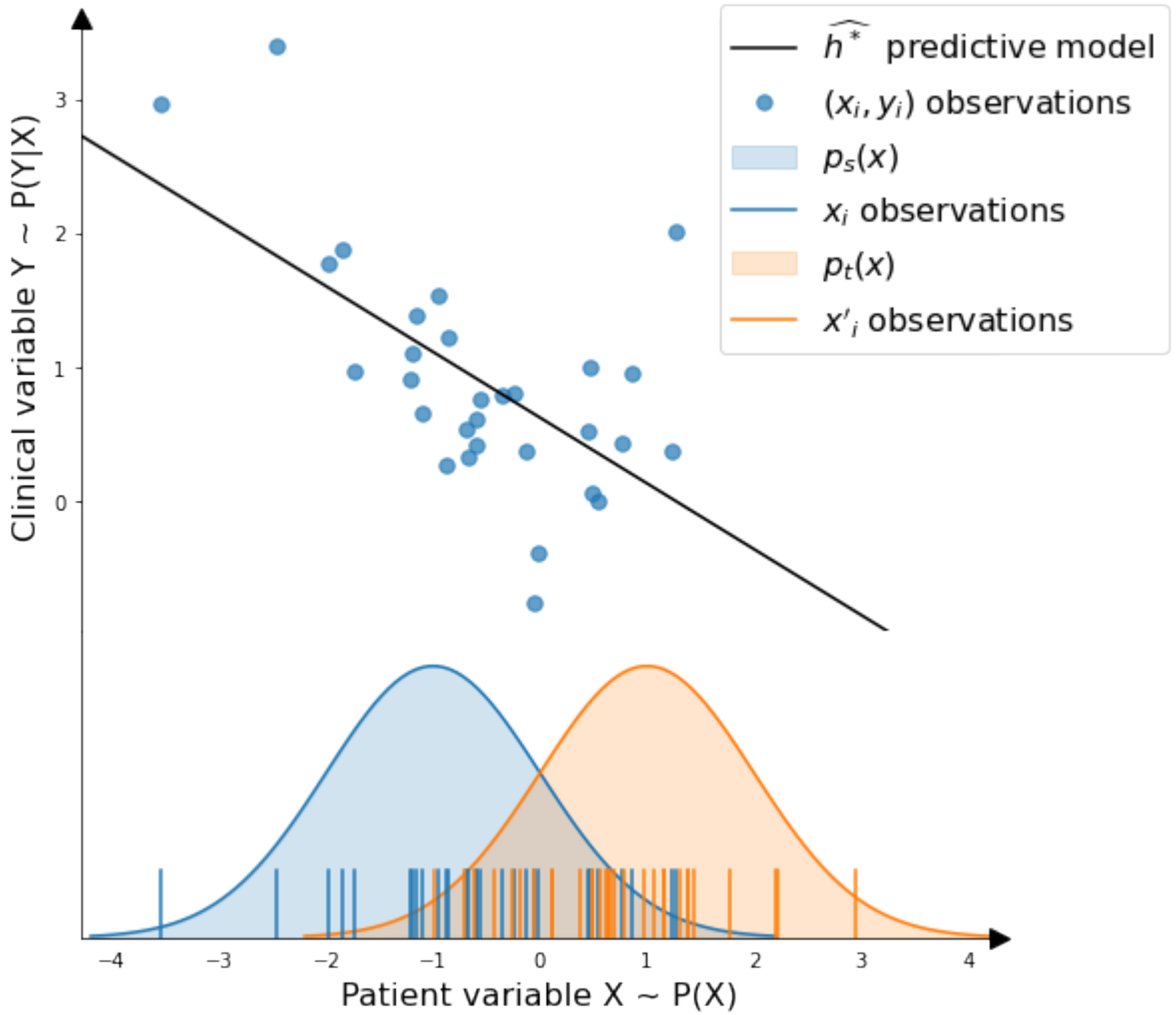


Figure – The learner has access to observations of the patient variable in the target domain.

Example

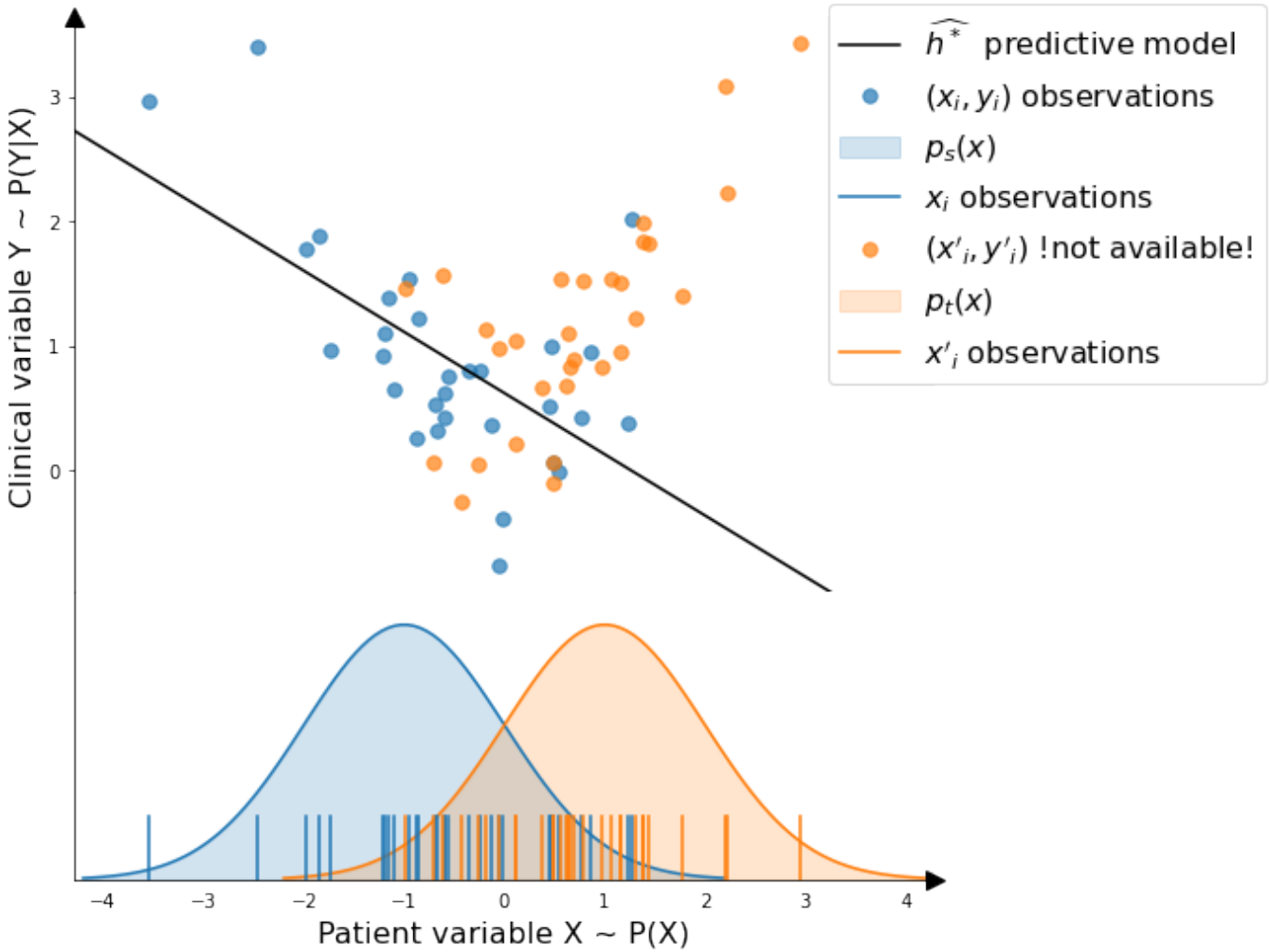


Figure – However, at training time, the learner has not access to target observations of the clinical variable. In this case, we plot them to see how the domain shift can mislead the learning of the model.

Example

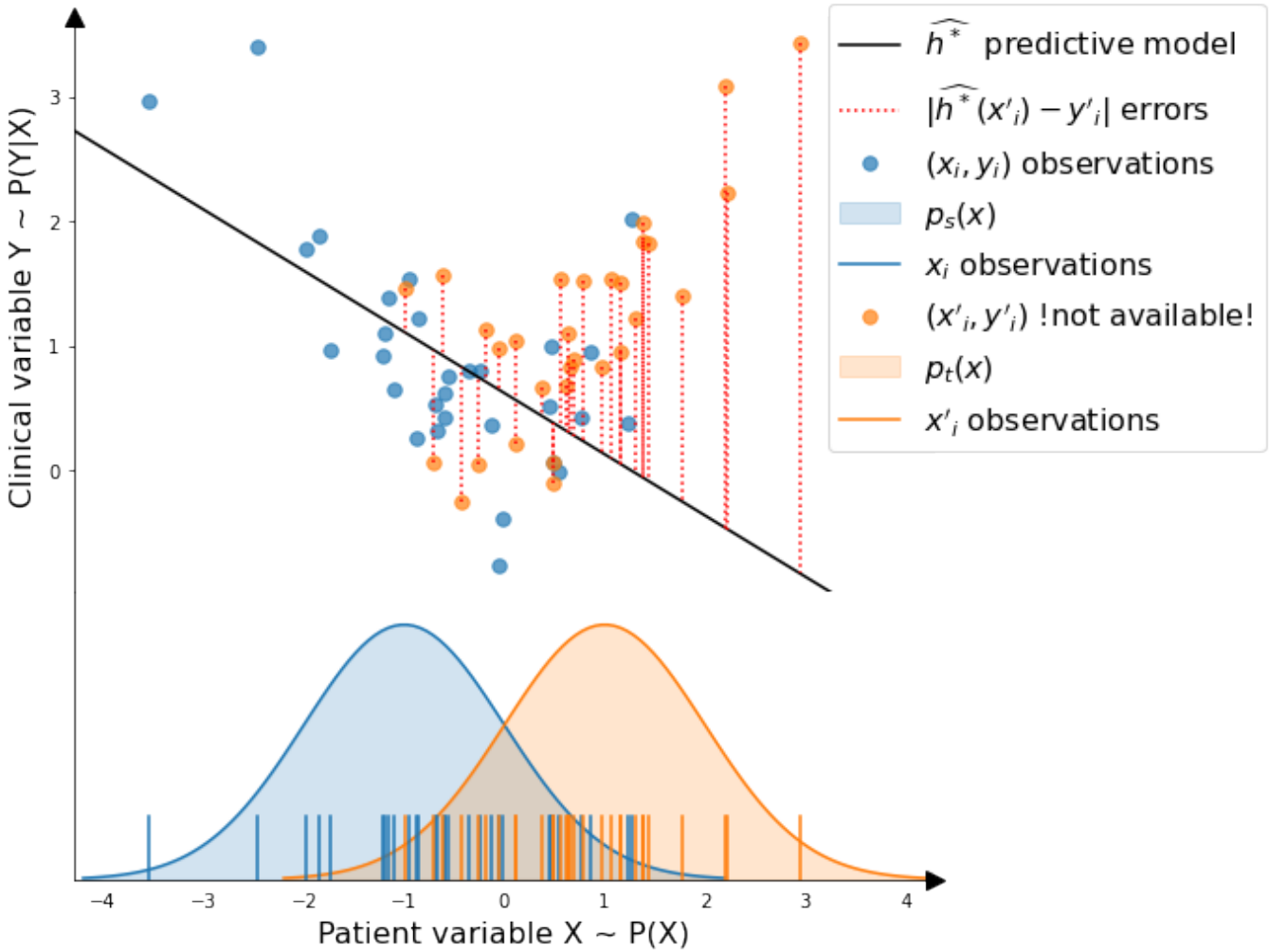


Figure – We can see that our model has been biased by the non representative source data and thus produce large errors on the target domain.

Notations

Let's introduce the following notations :

- **loss function** : $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ or $\ell : \mathcal{G} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.
  MSE : $\ell(y, y') = ||y - y||_2^2$
  CP : $\ell(C_\alpha, x, y) = \mathbf{1}(y \in C_\alpha(x))$
- **Hypothesis space** : $\mathcal{H}$, a functional space such that $h : \mathcal{X} \to \mathcal{Y}$ for any $h \in \mathcal{H}$.
  Example : $\mathcal{H} = \{h : x \to x\beta^T, \beta \in \mathbb{R}^p\}$ (linear hypotheses)

For any $h \in \mathcal{H}$, the target risk is defined as follows :

$$R_T(h) = \mathop{\mathbb{E}}_{(x,y)\sim P_T(X,Y)}[\ell(h(x), y)] \tag{1}$$

If target labels $y'_j \in \mathcal{Y}$ were available, a straightforward estimation of $R_T(h)$ would be :

$$\widehat{R_T}(h) = \frac{1}{n}\sum_{j=1}^{n} \ell(h(x'_j, y'_j)) \tag{2}$$

UDA Problem Formulation

UDA Problem

Let ℓ be a loss function and $\mathcal{H}$ a hypothesis space. We define the optimal target hypothesis $h^* \in \mathcal{H}$ as :

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \underset{(x,y) \sim P_T(X,Y)}{\mathbb{E}} [\ell(h(x), y)] \tag{3}$$

The goal of UDA is to find the best possible estimation of h^* based on the source labeled sample $\mathcal{S}$ and the target unlabeled sample $\mathcal{T}_{\mathcal{X}}$

The Importance Weighting Approach

Assuming $P_S(X, Y), P_T(X, Y)$ have density functions $p_s(x, y), p_t(x, y)$ and $w(x, y) = p_t(x, y)/p_s(x, y)$:

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim P_T(X,Y)}[\ell(h(x), y)] &= \int_{\mathcal{X}\times\mathcal{Y}} p_t(x, y)\, \ell(h(x), y) dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \frac{p_t(x, y)}{p_s(x, y)} p_s(x, y)\, \ell(h(x), y) dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} w(x, y) p_s(x, y)\, \ell(h(x), y) dxdy \\
&= \mathop{\mathbb{E}}_{(x,y)\sim P_S(X,Y)}[w(x, y)\ell(h(x), y)]
\end{aligned}$$

The target risk can be written as a "reweighted" source risk :

$$\mathop{\mathbb{E}}_{(x,y)\sim P_T(X,Y)}[\ell(h(x), y)] = \mathop{\mathbb{E}}_{(x,y)\sim P_S(X,Y)}[w(x, y)\ell(h(x), y)]$$

Density ratio

Definition 2 (Density Ratio)

Let be $p_s(x, y)$ and $p_t(x, y)$ the respective source and target density functions such that $\text{supp}(p_T(x, y)) \subset \text{supp}(p_S(x, y))$. The **density ratio** or **importance weight** is defined, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, as :

$$w(x, y) = \frac{p_t(x, y)}{p_s(x, y)}$$

Remark : the support of the target distribution is required to be included in the support of the source distribution, i.e :

$$p_s(x, y) = 0 \implies p_t(x, y) = 0, \quad \forall (x, y) \in \mathcal{X} \times \mathcal{Y}$$

Problem : $w(x, y)$ involves $p_t(x, y) = p_t(y|x)p_t(x)$ which requires target labels to be estimated.

Covariate-Shift Assumption

Definition 3 (Covariate Shift)

The source and target joint distributions $P_S(X, Y)$ and $P_T(X, Y)$ follows the **covariate-shift** assumption if, for any $x \in \mathcal{X}$:

$$P_T(Y|X = x) = P_S(Y|X = x) \tag{4}$$

Proposition 1

*Let $p_s(x, y)$ and $p_t(x, y)$ be the source and target density functions. If the **covariate-shift** assumption holds then, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the **density ratio** can be written :*

$$w(x, y) = w(x) = \frac{p_t(x)}{p_s(x)} \tag{5}$$

Proof : $w(x, y) = \frac{p_t(x,y)}{p_s(x,y)} = \frac{p_t(y|x)p_t(x)}{p_s(y|x)p_s(x)} = \frac{p_t(x)}{p_s(x)}$

Covariate-Shift Examples

Definition 4 (Covariate Shift)

The source and target joint distributions $P_S(X, Y)$ and $P_T(X, Y)$ follows the **covariate-shift** assumption if, for any $x \in \mathcal{X}$:

$$P_T(Y|X = x) = P_S(Y|X = x) \tag{6}$$

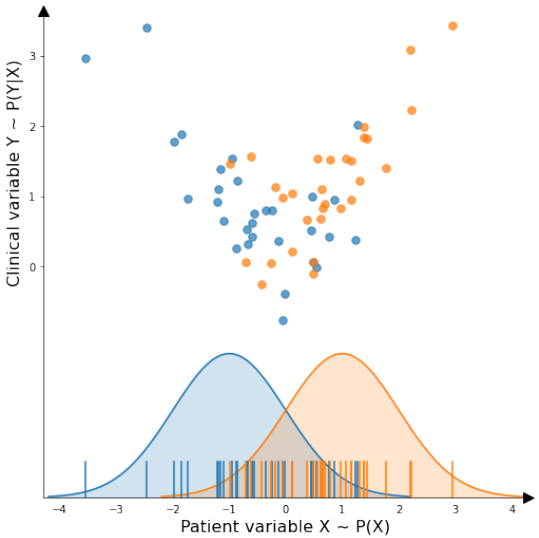


Figure – (A)

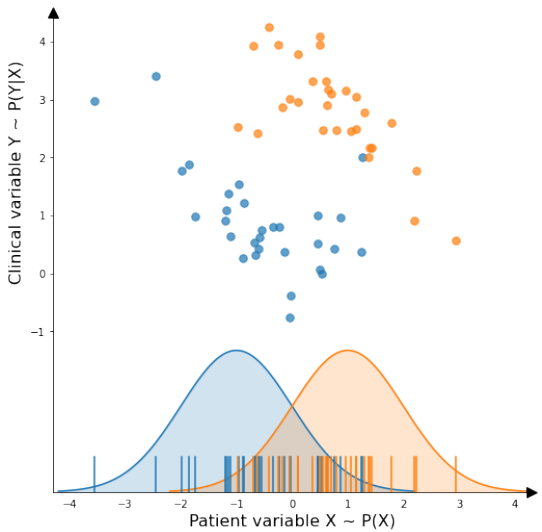


Figure – (B)

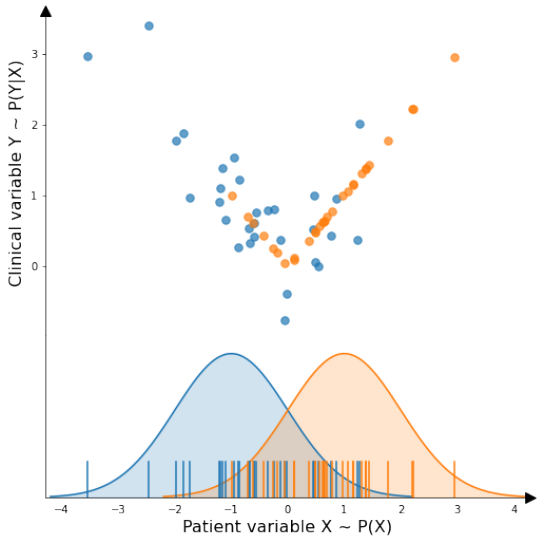


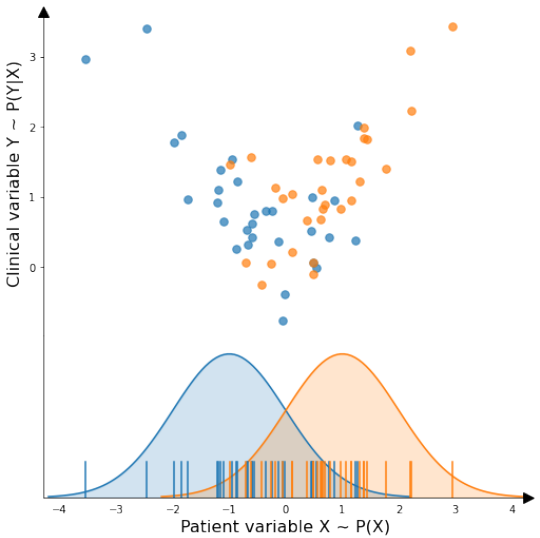
Figure – (C)

Covariate-Shift Examples

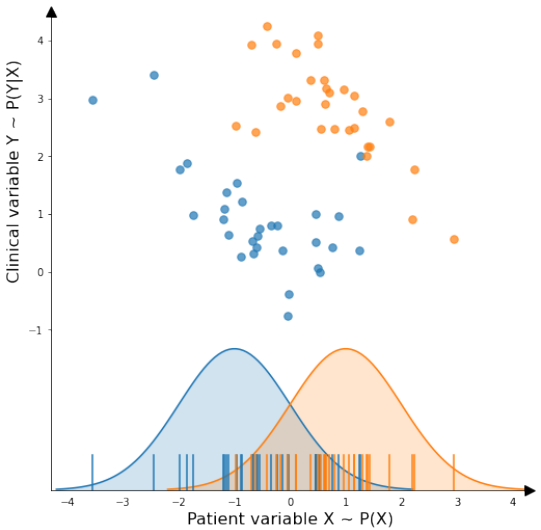
Definition 5 (Covariate Shift)

The source and target joint distributions $P_S(X, Y)$ and $P_T(X, Y)$ follows the **covariate-shift** assumption if, for any $x \in \mathcal{X}$:

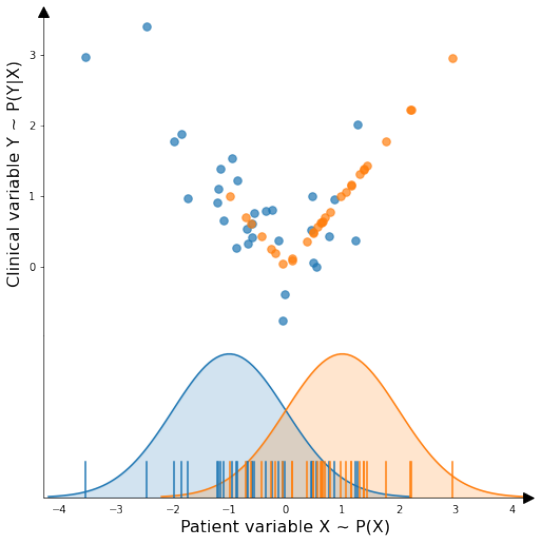
$$P_T(Y|X = x) = P_S(Y|X = x) \tag{7}$$



$P_S(Y|x) \sim \mathcal{N}(|x|, 1)$
$P_T(Y|x) \sim \mathcal{N}(|x|, 1)$



$P_S(Y|x) \sim \mathcal{N}(|x|, 1)$
$P_T(Y|x) \sim \mathcal{N}(4 - |x|, 1)$



$P_S(Y|x) \sim \mathcal{N}(|x|, 1)$
$P_T(Y|x) \sim \mathcal{N}(|x|, 0.01)$

Covariate-Shift in real life

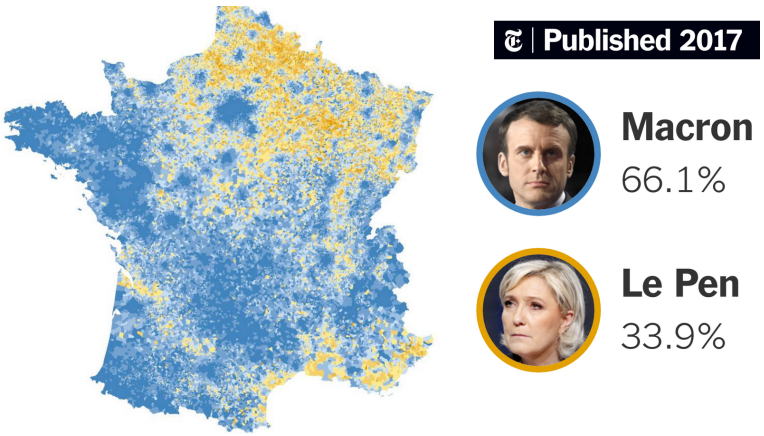


Figure – **Election Polls** (source : nytimes.com). It is generally assumed that people give the same vote for the survey and the election.

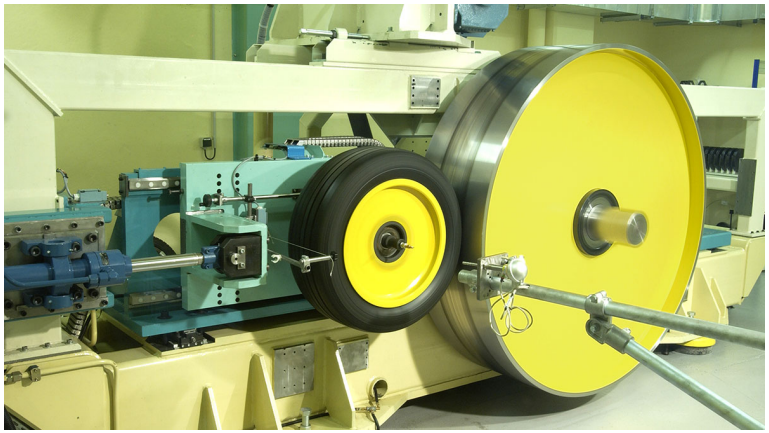


Figure – **Industrial Predictive Models** (source : michelin.com). The measuring machine used to build the training set is supposed to be the same than the one used for the target set (ex : they have the same measurement noise)

Covariate-Shift and Support Overlap Assumption

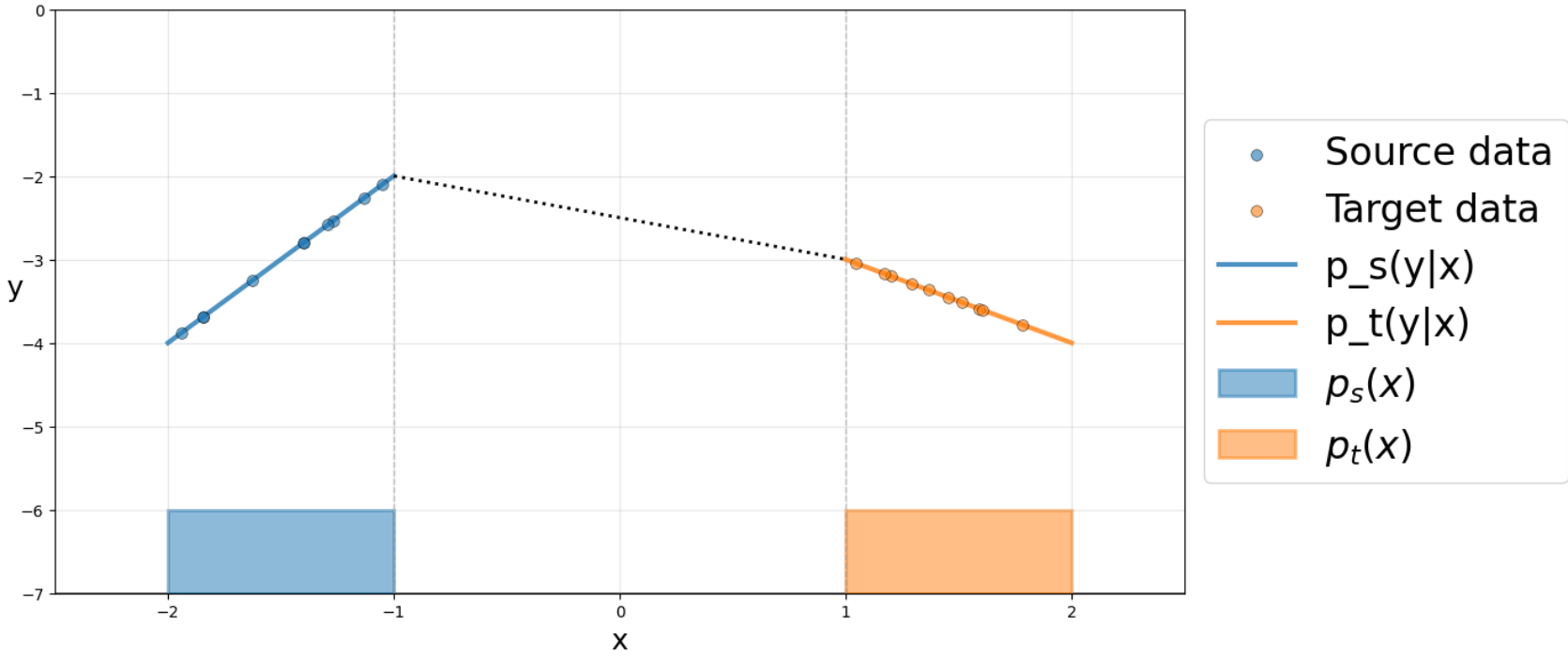


Figure – The covariate shift assumption is meaningful only when the source and target distributions $p_s(x)$ and $p_t(x)$ have overlapping support. Without such overlap, the conditional distribution $p(y \mid x)$ can always be defined identically on both domains.

Instance-based approach

We have shown that, for any $h \in \mathcal{H}$:

$$\mathop{\mathbb{E}}_{(x,y)\sim P_T(X,Y)}[\ell(h(x),y)] = \mathop{\mathbb{E}}_{(x,y)\sim P_S(X,Y)}[w(x,y)\ell(h(x),y)]$$

Under the **covariate-shift** assumption, we have :

$$\mathop{\mathbb{E}}_{(x,y)\sim P_T(X,Y)}[\ell(h(x),y)] = \mathop{\mathbb{E}}_{(x,y)\sim P_S(X,Y)}[w(x)\ell(h(x),y)]$$

$\Longrightarrow$ Assuming that $w(x)$ is known for any $x \in \mathcal{X}$, the target risk of $h \in \mathcal{H}$ can be estimated thanks to the source sample $S = \{(x_1, y_1), ..., (x_m, y_m)\}$:

$$\mathop{\mathbb{E}}_{(x,y)\sim P_T(X,Y)}[\ell(h(x),y)] \approx \frac{1}{m}\sum_{i=1}^{m} w(x_i)\ell(h(x_i),y_i)$$

The Importance Weighting Approach

We have shown that, for any $h \in \mathcal{H}$:

$$\mathop{\mathbb{E}}_{(x,y)\sim P_T(X,Y)}[\ell(h(x),y)] \approx \frac{1}{m}\sum_{i=1}^{m} w(x_i)\ell(h(x_i),y_i)$$

Moreover, the quantity $w(x_i) = p_t(x_i)/p_s(x_i)$ does not involve labels and can be estimated using only the source and target input samples $S_{\mathcal{X}} = \{x_1, ..., x_m\} \sim P_S(X)$, $T = \{x'_1, ..., x'_m\} \sim P_T(X)$

UDA Instance-based

A UDA instance-based approach consist in solving the following optimization problem :

$$\text{Estimating : } \quad \widehat{w}(x_i) \quad \forall\, x_i \in S_{\mathcal{X}}$$

$$\widehat{h^*} = \mathop{\text{argmin}}_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \widehat{w}(x_i)\ell(h(x_i),y_i)$$

Exercise : Weighted Regression

Setup

Let's consider the regression problem with $X \in \mathbb{R}^{m \times p}$ the matrix of input data $\{x_1, ..., x_m\}$ such that $x_i \in \mathbb{R}^p$ is the i^{th} row of X and $Y \in \mathbb{R}^{m \times 1}$ the vector of output data $\{y_1, ..., y_m\}$ with $y_i \in \mathbb{R}$ for any $i \in [|1, m|]$.
We consider the UDA instance-based problem where $w_i = p_t(x_i)/p_s(x_i)$ is assumed to be known for any x_i. The loss function is the squared error : $\ell(y, y') = (y - y')^2$ and $\mathcal{H}$ the hypothesis space of linear hypotheses $\mathcal{H} = \{h : x \to x\beta^T, \beta \in \mathbb{R}^p\}$

Question :

Find a close-form solution β^* of the UDA instance-based problem involving X, Y and $\Delta = \text{diag}(\sqrt{w_1}, ..., \sqrt{w_m})$.

Exercise : Weighted Regression

Solution :

The problem can be formulated as follows :

$$\beta^* = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{m} w_i (x_i \beta^T - y_i)^2$$

By denoting $\Delta = \operatorname{diag}(\sqrt{w_1}, ..., \sqrt{w_m})$, we have :

$$\begin{aligned} \beta^* &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} ||\Delta(X\beta^T - Y)||_2^2 \\ &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} ||\Delta X\beta^T - \Delta Y||_2^2 \end{aligned}$$

Assuming $X^T \Delta^2 X$ invertible, we have :

$$\begin{aligned} \beta^* &= [(\Delta X)^T \Delta X]^{-1} (\Delta X)^T \Delta Y \\ &= [X^T \Delta^2 X]^{-1} X^T \Delta^2 Y \end{aligned}$$

Example

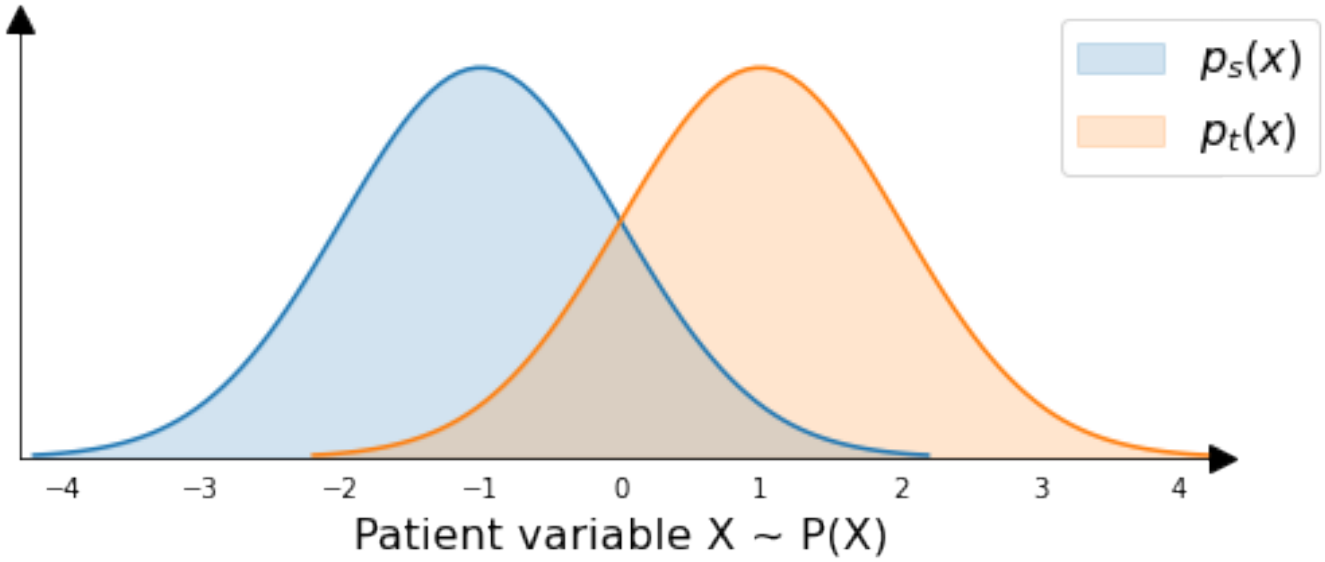


Figure – Let's consider the two shifted Gaussians problem introduced previously.

Example

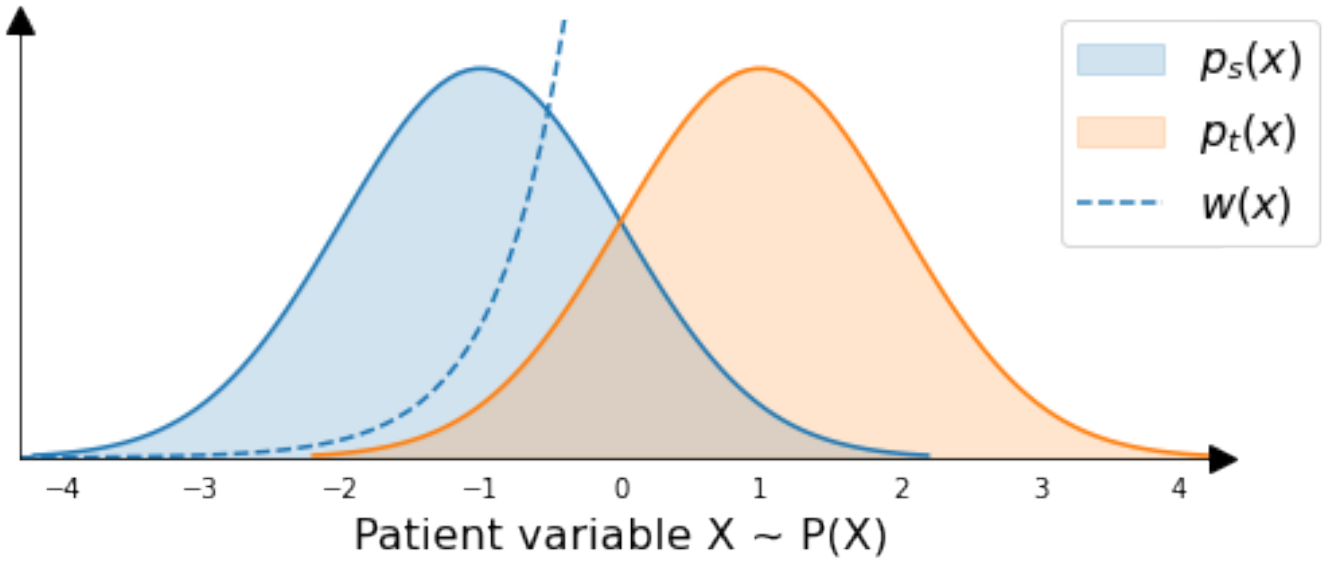


Figure – If the density functions of both domains are known, we can derive the density ratio $w(x) = p_t(x)/p_s(x)$ (here in dashed line).

Example

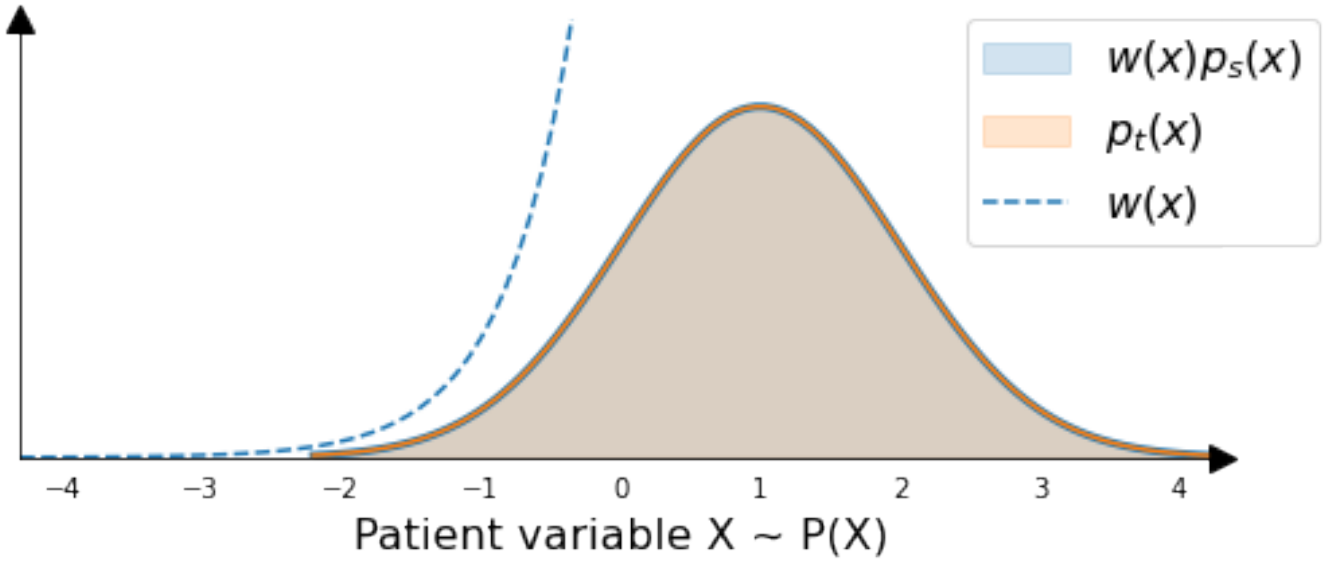


Figure – We can see that the target density to the source reweighted density match perfectly.

Example

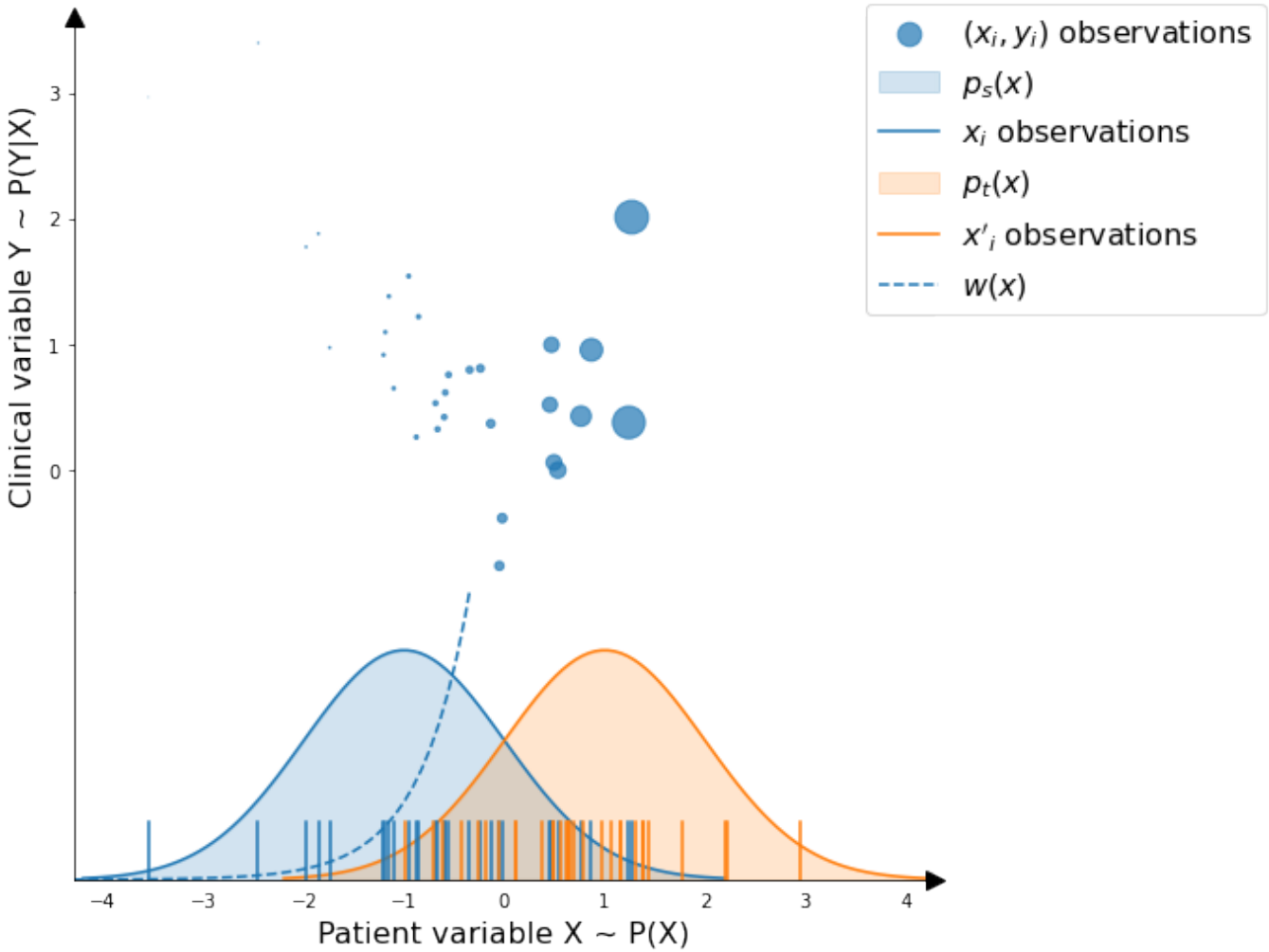


Figure – We display here the source obsrevations (x_i, y_i) sized with their corresponding weights $w(x_i)$ (the larger $w(x_i)$, the larger the size of the blue dots).

Example

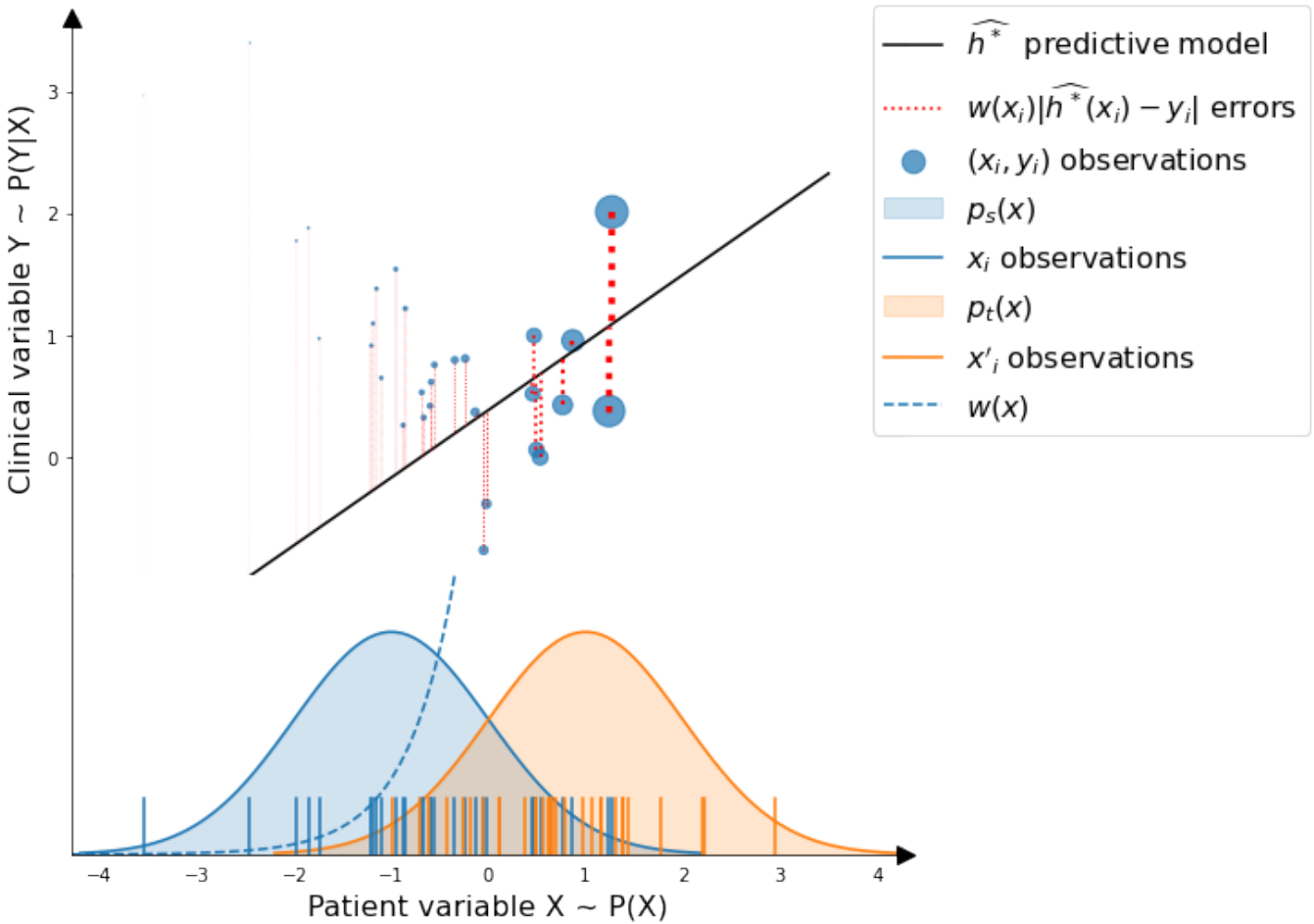


Figure – We now fit a linear model with the reweighted objective.

Example

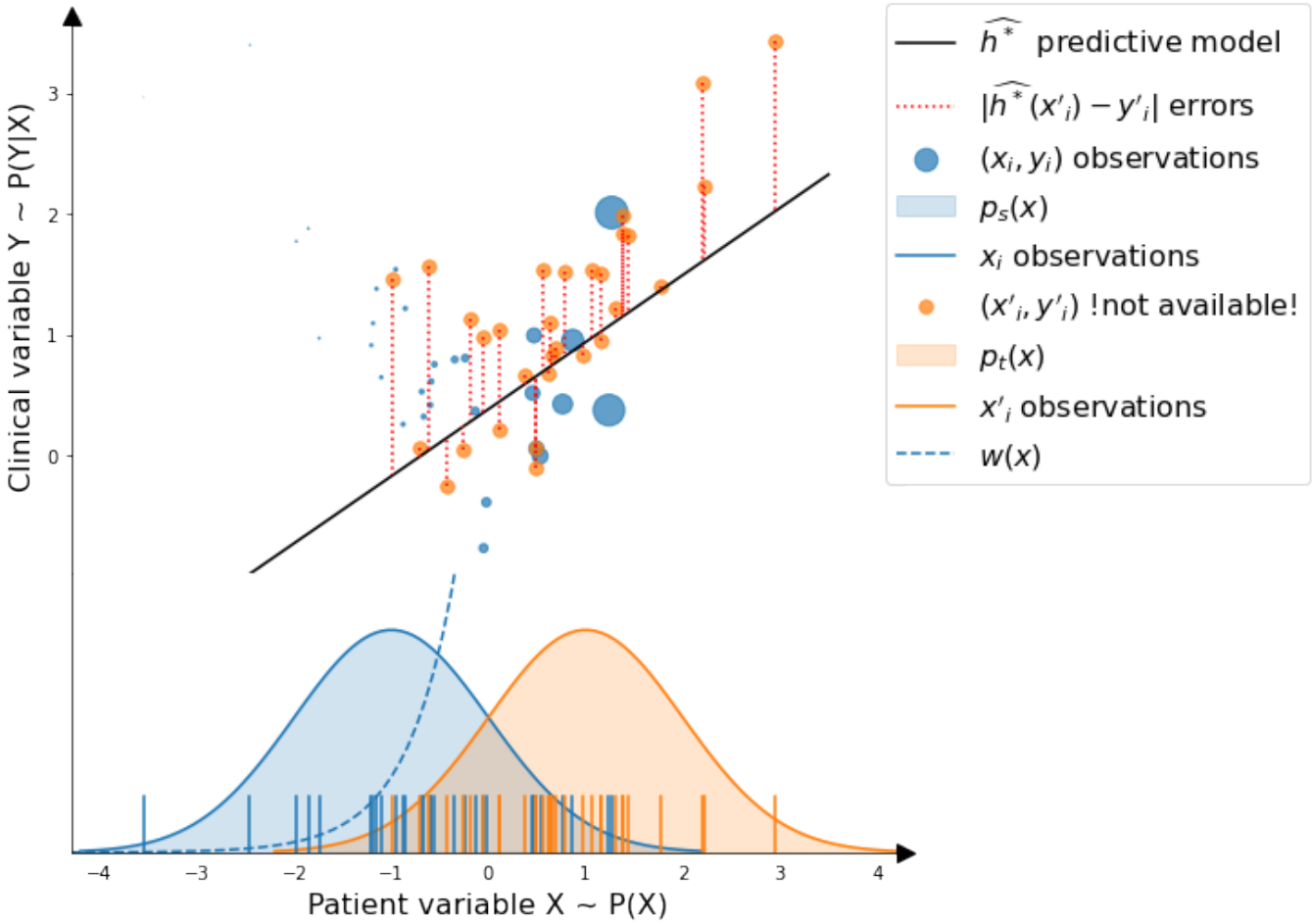


Figure – We observe that the linear model learned using this procedure provides a better fit to the target data.

Example

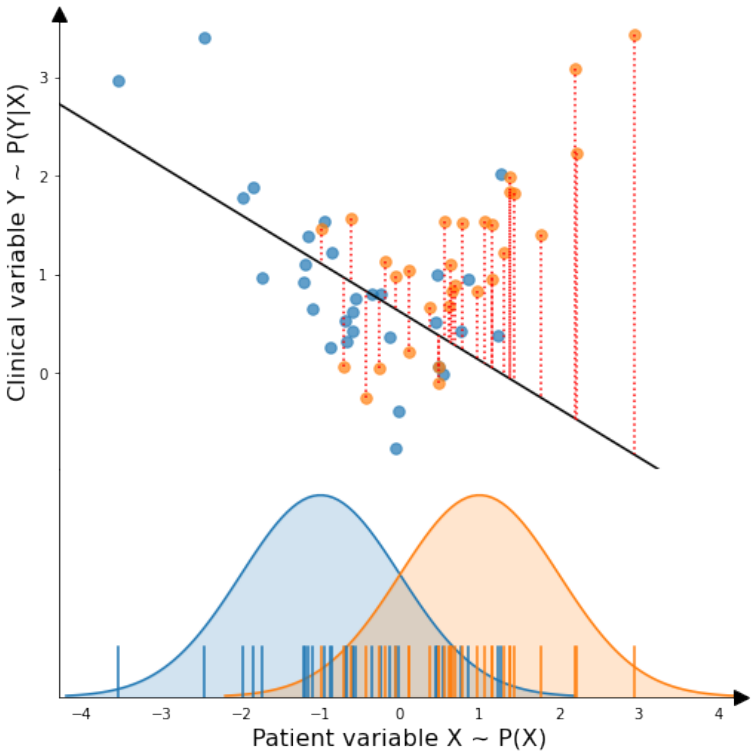


Figure – Without Importance Weighting

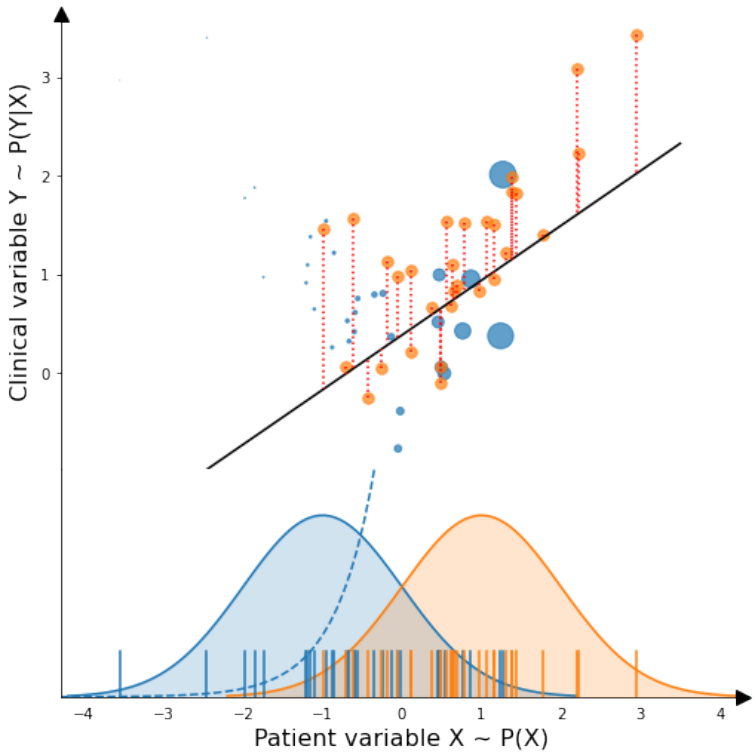


Figure – With Importance Weighting

Density Ratio Estimation

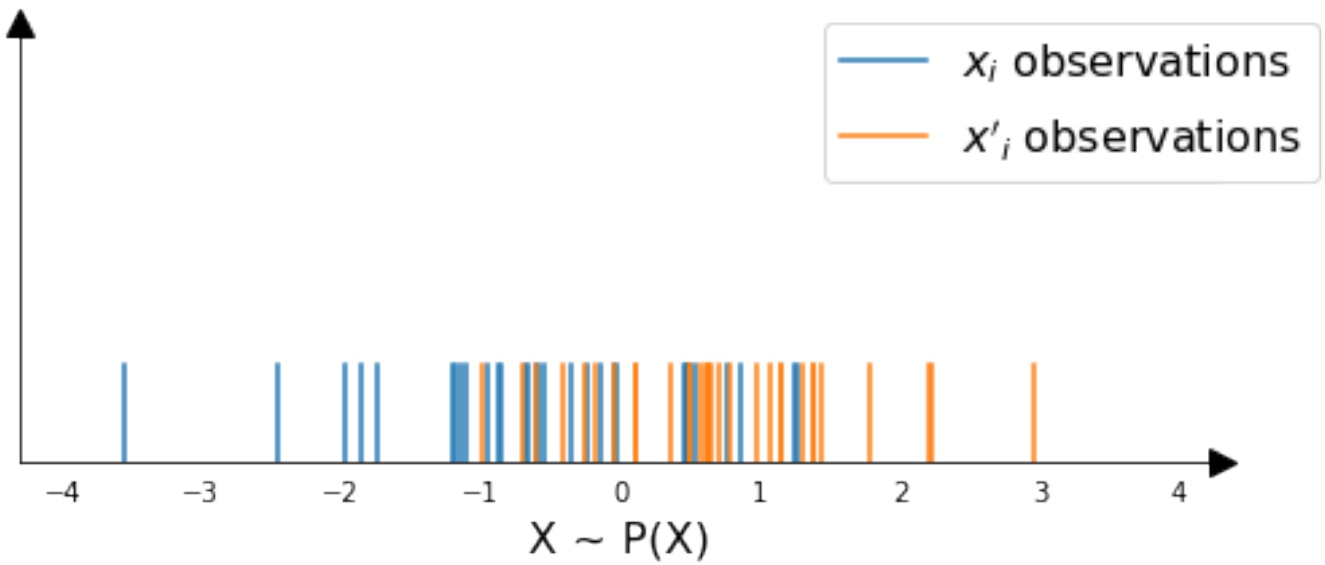


Figure – Generally, $p_s(x)$ and/or $p_t(x)$ are not known. The learner has only access to samples $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{T}_{\mathcal{X}}$ respectively drawn according to $P_S(X)$ and $P_T(X)$.

Problem : How can we estimate $w(x)$ based on $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{T}_{\mathcal{X}}$?

Density Ratio Estimation

Multiple approaches exist to estimate $w(x)$ and can be summarized as follows :

Density Estimation :

- Parametric Density Estimation
- Kernel Density Estimation

Heuristic :

- Nearest Neighbors Weighting [Loog, 2012]

Distribution Matching :

- Kullback–Leibler Importance Estimation Procedure (KLIEP) [Sugiyama et al., 2007]
- Kernel Mean Matching [Huang et al., 2007]

Domain Classifier

- Linear/Kernel Classifier [Bickel et al., 2007]
- Neural Network Classifier [Zhang et al., 2018]

Density Estimation

Density Estimation approaches consist in estimating $p_s(x)$, $p_t(x)$ based on the respective samples $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{T}_{\mathcal{X}}$. Then, the density ratio is computed for any $x_i \in \mathcal{S}_{\mathcal{X}}$ as follows :

$$\widehat{w}(x_i) = \frac{\widehat{p}_t(x_i)}{\widehat{p_s}(x_i)}$$

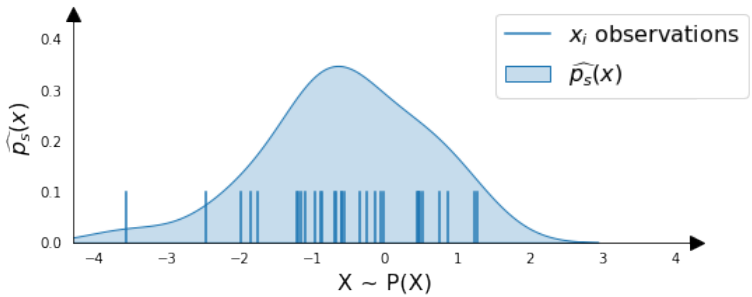


Figure – 1D Density Estimation

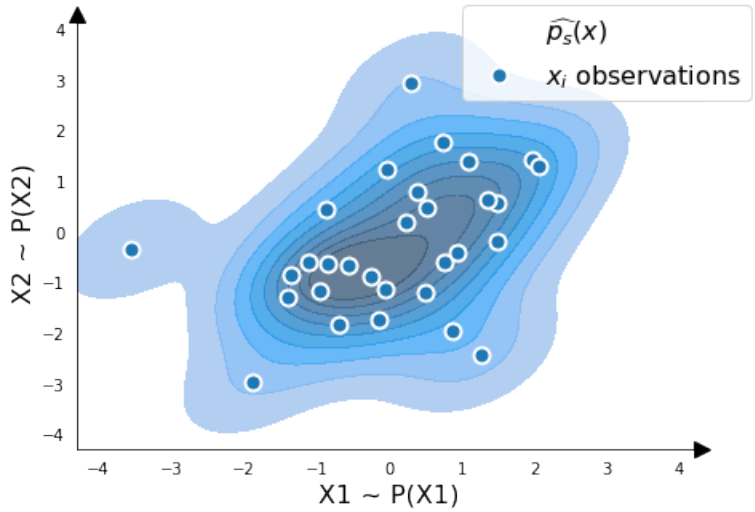


Figure – 2D Density Estimation

Parametric Density Estimation

The learner assumes a prior input distribution with densities $p_s(\theta_s, x)$, $p_t(\theta_t, x)$ for the source and target domains respectively. The parameters θ_s and θ_t of the distributions are then estimated through Maximum-Likelihood estimation.

$$\widehat{\theta_s^*} = \underset{\theta_s \in \Theta_s}{\operatorname{argmin}} \sum_{x_i \in \mathcal{S}_X} \log(p_s(\theta_s, x_i))$$

$$\widehat{\theta_t^*} = \underset{\theta_t \in \Theta_t}{\operatorname{argmin}} \sum_{x_i' \in \mathcal{T}_X} \log(p_t(\theta_t, x_i'))$$

$$\widehat{w}(x_i) = \frac{p_t(\widehat{\theta_t^*}, x_i)}{p_s(\widehat{\theta_s^*}, x_i)}$$

Parametric Density Estimation

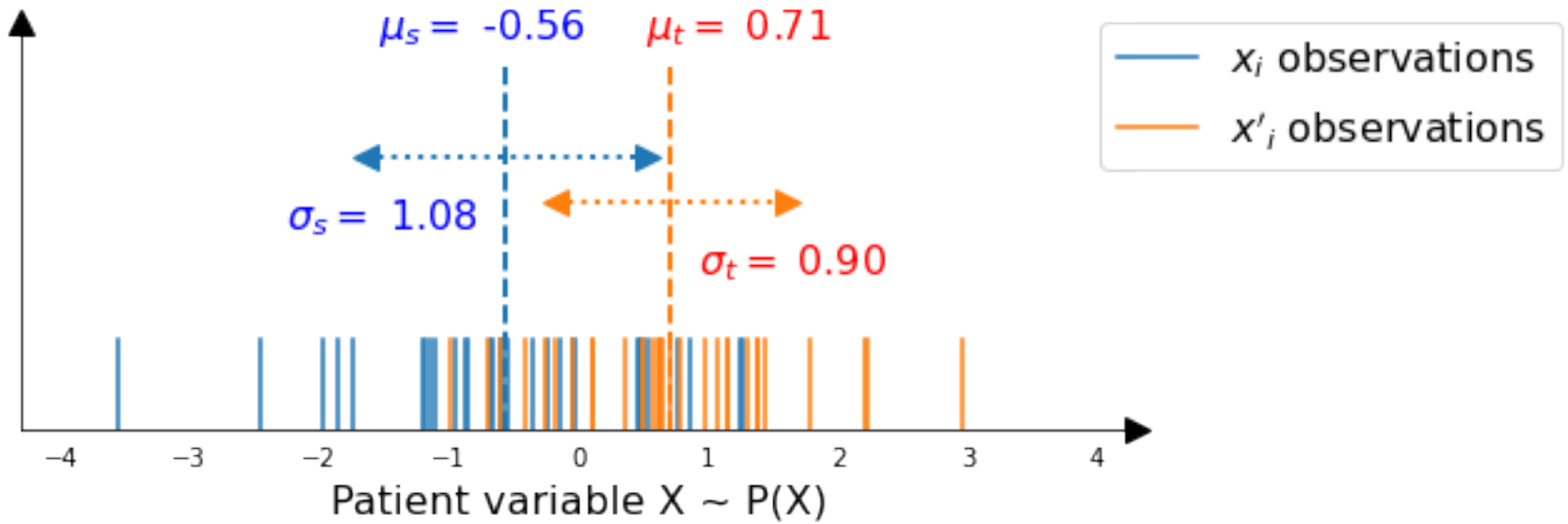


Figure – To perform density estimation with a Gaussian prior, the learner estimates the mean and standard deviations of both distributions based on $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{T}_{\mathcal{X}}$.

Parametric Density Estimation

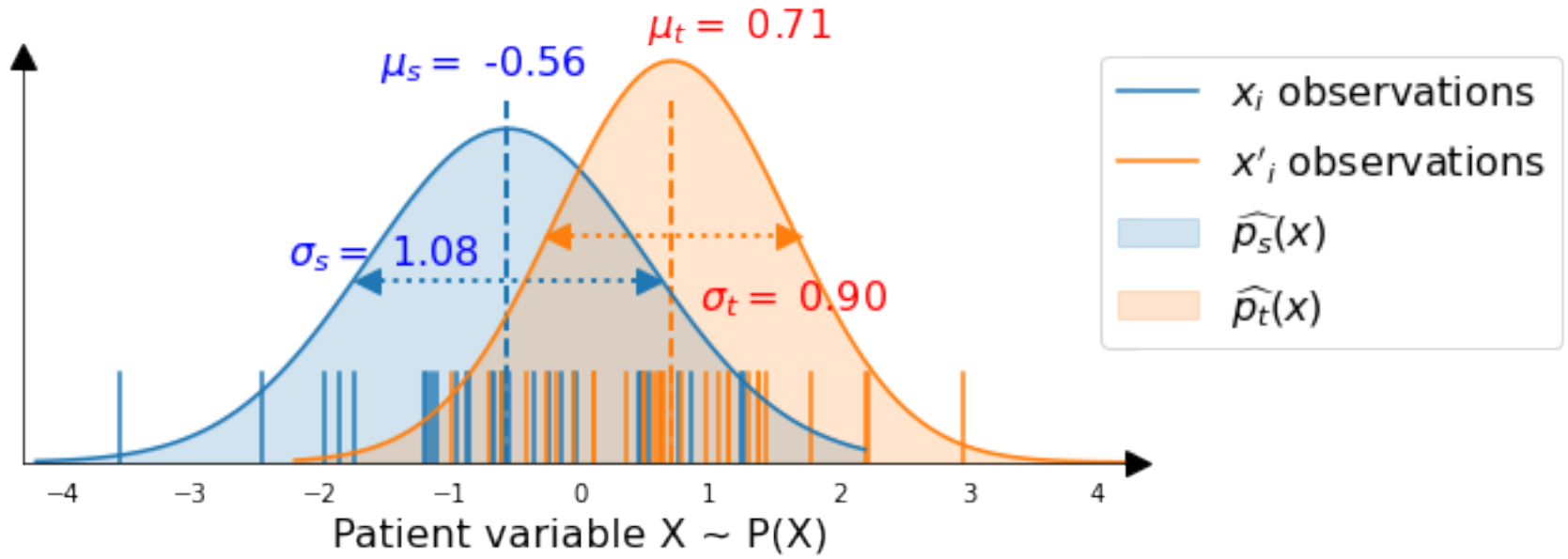


Figure – To perform density estimation with a Gaussian prior, the learner estimates the mean and standard deviations of both distributions based on $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{T}_{\mathcal{X}}$.

Kernel Density Estimation

Let's consider $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ a kernel function. The learner computes estimations of the densities $p_s(x)$, $p_t(x)$ as sums of kernels centered in each observation $x_i \in \mathcal{S}_\mathcal{X}$ (resp $x'_i \in \mathcal{T}_\mathcal{X}$).

$$\widehat{p_s}(x) = C_s \sum_{x_i \in \mathcal{S}_\mathcal{X}} k(x, x_i)$$

$$\widehat{p_t}(x) = C_t \sum_{x'_i \in \mathcal{T}_\mathcal{X}} k(x, x'_i)$$

$$\widehat{w}(x_i) = \frac{\widehat{p_t}(x_i)}{\widehat{p_s}(x_i)}$$

Remark : A typical kernel function is the Gaussian kernel : $k : (x, x') \to \exp(-\frac{||x-x'||_2^2}{2\sigma^2})$. $\sigma \in \mathbb{R}_+^*$ is the kernel bandwidth and can be chosen through Maximum Likelihood estimation.

Kernel Density Estimation

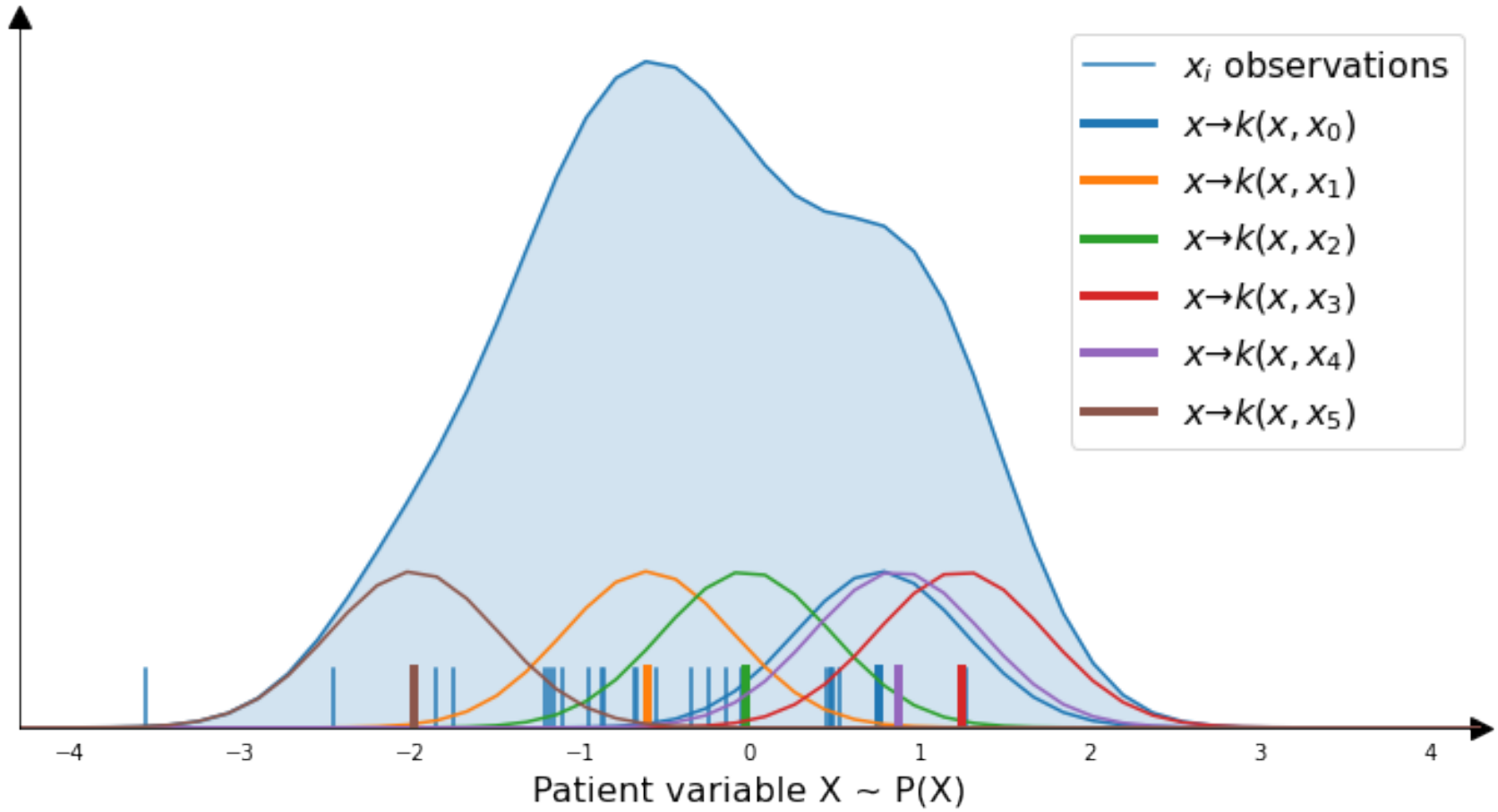


Figure – Kernel Density Estimation consist in aggregating multiple densities functions centered in each observation.

Nearest Neighbors Weighting

Let's consider $K \in \mathbb{N}^*$ and define $\mathcal{V}_K(x'_j)$ as the set of K-Nearest-Neighbors of $x'_j \in \mathcal{T}_\mathcal{X}$ in $\mathcal{S}_\mathcal{X}$.

The density ratio $\widehat{w}(x_i)$ is directly estimated through the following heuristic :

$$\widehat{w}(x_i) = \text{Card}\left(\left\{x'_i \in \mathcal{T}_\mathcal{X} \mid x_i \in \mathcal{V}_K(x'_j)\right\}\right)$$

Remark : if $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is a distance over $\mathcal{X}$, a formal definition of $\mathcal{V}_K(x'_j)$ can be written as follows :

$$\mathcal{V}_K(x'_j) = \left\{ x \in \mathcal{S}_\mathcal{X}; \sum_{x_i \in \mathcal{S}_\mathcal{X}} \mathbb{1}\left(d(x_i, x'_j) < d(x, x'_j)\right) < K \right\}$$

Nearest Neighbors Weighting

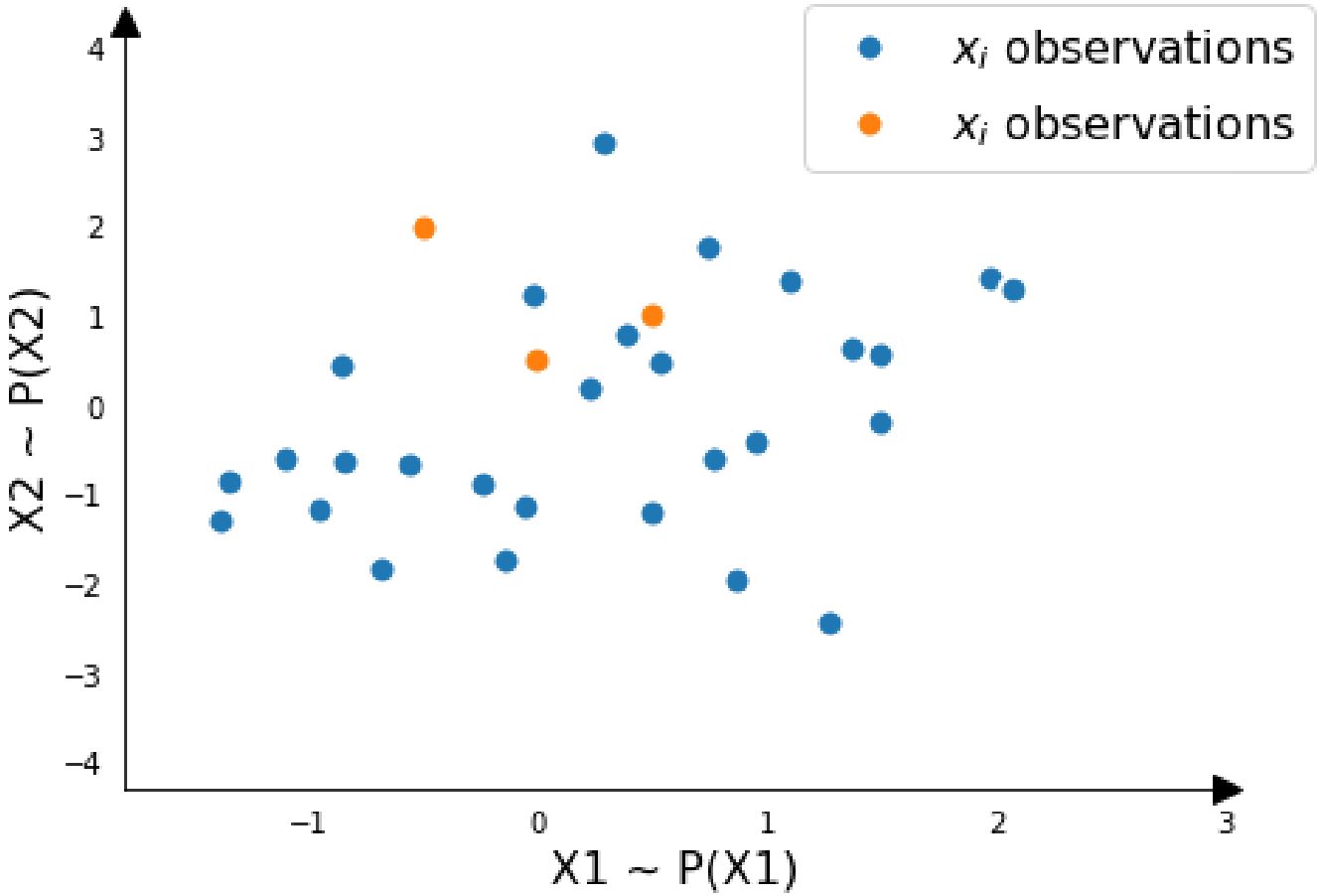


Figure – Let's consider a source and target input samples drawn according to different distributions in $\mathbb{R}^2$

Nearest Neighbors Weighting

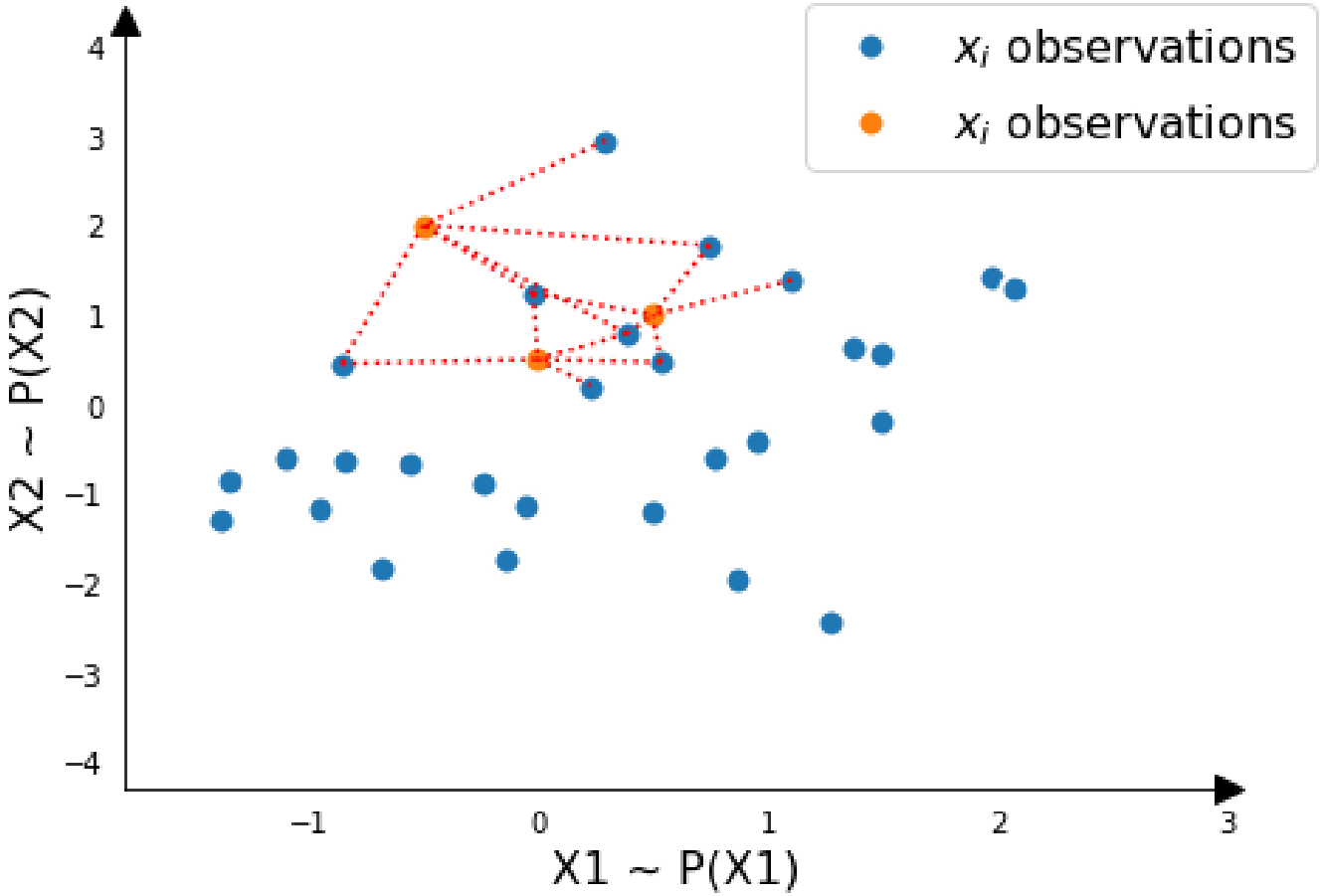


Figure – The Nearest Neighbors Weighting first computes the distance from each target data (orange) to every source data (blue). Then, it selects the K closest source data for each target (here $K = 5$).

Nearest Neighbors Weighting

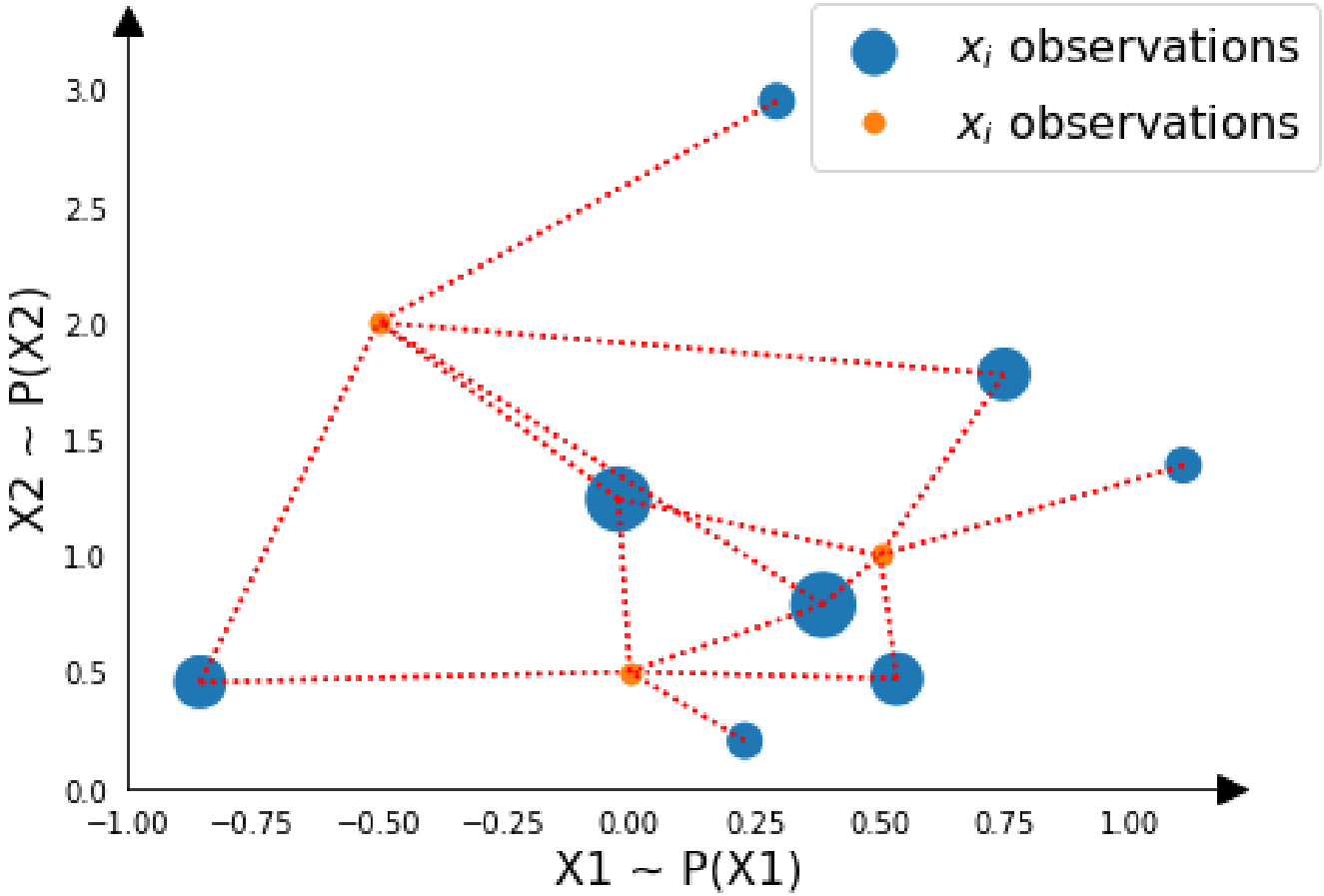


Figure – Finally, each source data is reweighted according to the number of times it has been selected as nearest neighbor by a target data.

Distribution Matching

Distribution Matching

Distribution Matching approaches consist in directly estimating the density ratio by looking for the weighting $w(x)$ that minimizes a "discrepancy metric" between $w(x)p_s(x)$ and $p_t(x)$:

$$w^* = \underset{w}{\operatorname{argmin}}\, D(p_t(x), w(x)p_s(x))$$

With D a metric measuring the "closeness" of two distributions.

Problematic : How to measure distribution "closeness" ?

Distribution Matching

Problematic : How to measure distribution "closeness" ?

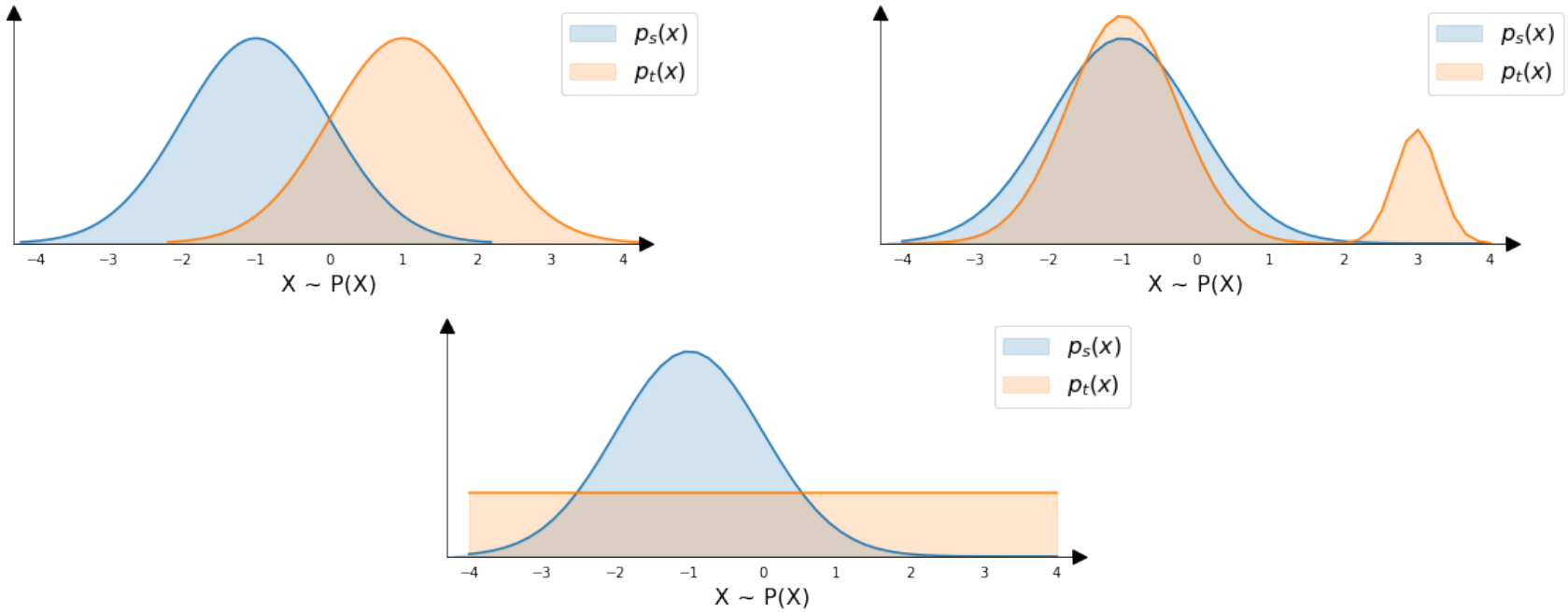


Figure – Which pair of distributions are the closest ?

Distribution Matching

Problematic : How to measure distribution "closeness" ?

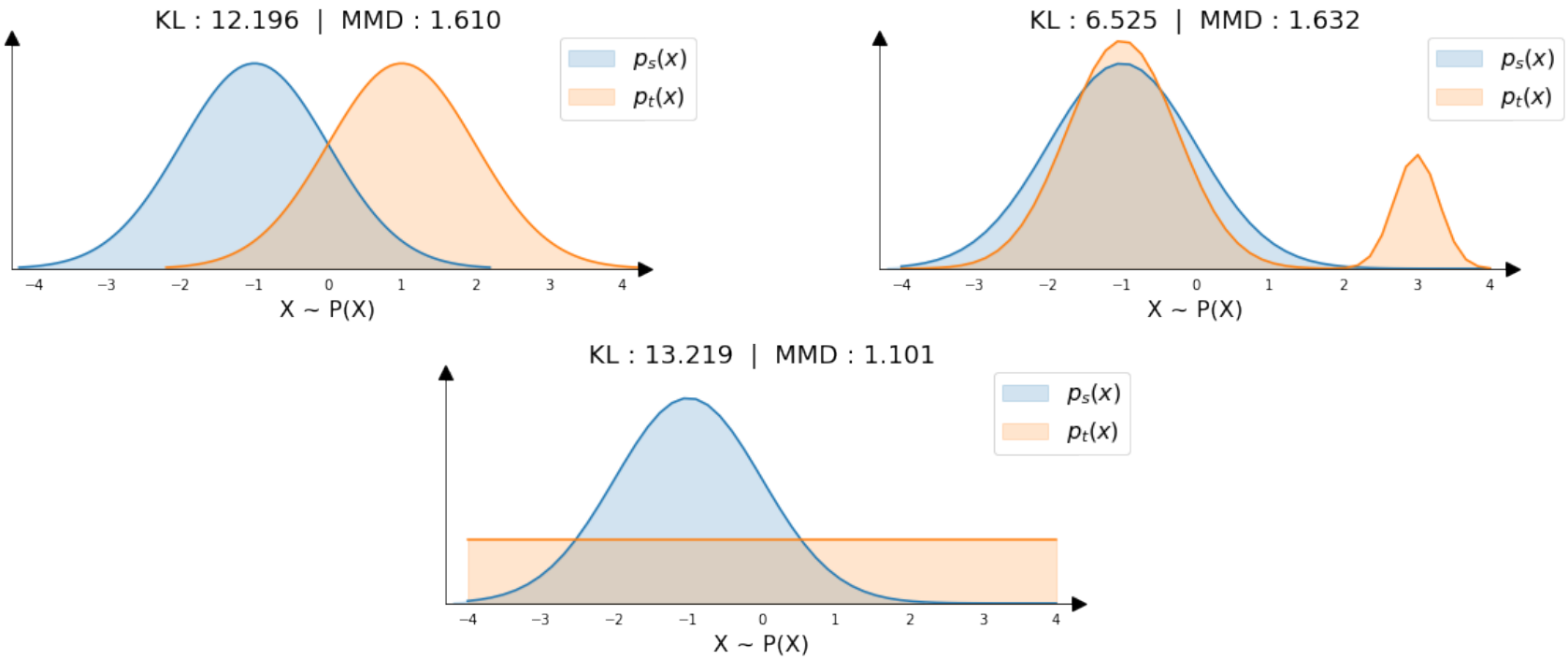


Figure – The notion of "distribution closeness" depends on the considered metric.

Examples of Divergences Between Distributions

Total Variation (TV)

$$\mathrm{TV}(p_t, p_s^w) = \frac{1}{2} \int_{\mathcal{X}} \left| p_t(x) - p_s^w(x) \right| dx.$$

Kullback–Leibler Divergence (KL)

$$\mathrm{KL}(p_t, p_s^w) = \int_{\mathcal{X}} p_t(x) \log\left(\frac{p_t(x)}{p_s^w(x)} \right) dx.$$

Maximum Mean Discrepancy (MMD)

Let $\phi : \mathcal{X} \to H$ be the feature map into an RKHS H. Then

$$\mathrm{MMD}(p_t, p_s^w) = \left\| \int_{\mathcal{X}} \left(p_t(x) - p_s^w(x) \right) \phi(x) \, dx \right\|_H.$$

Empirical Divergences Between Distributions

Empirical KL Divergence

$$\widehat{\mathrm{KL}}(p_t, p_s^w) = \frac{1}{n} \sum_{x' \in \mathcal{T}_\mathcal{X}} \log\left(\frac{p_t(x')}{p_s^w(x')}\right)$$
$$= -\frac{1}{n} \sum_{x' \in \mathcal{T}_\mathcal{X}} \log\left(w(x)\right) + \frac{1}{n} \sum_{x' \in \mathcal{T}_\mathcal{X}} \log\left(p_t(x)/p_s(x)\right).$$

Empirical MMD

Let $k(\cdot,\cdot)$ be such that $k(x,x') = \langle \phi(x), \phi(x') \rangle_\mathcal{H}$ and $\sum_{x \in \mathcal{S}_\mathcal{X}} w(x) = m$. Then

$$\widehat{\mathrm{MMD}}(p_t, p_s^w)^2 = \left\| \frac{1}{n} \sum_{x' \in \mathcal{T}_\mathcal{X}} \phi(x') - \frac{1}{m} \sum_{x \in \mathcal{S}_\mathcal{X}} w(x)\, \phi(x) \right\|_\mathcal{H}^2$$
$$= \frac{1}{n^2} \sum_{x,x' \in \mathcal{T}_\mathcal{X}} k(x,x') - \frac{2}{nm} \sum_{x' \in \mathcal{T}_\mathcal{X}} \sum_{x \in \mathcal{S}_\mathcal{X}} w(x)\, k(x,x')$$
$$+ \frac{1}{m^2} \sum_{x,x' \in \mathcal{S}_\mathcal{X}} w(x) w(x')\, k(x,x').$$

Kullback–Leibler Importance Estimation Procedure (KLIEP)

The KLIEP algorithm looks for the density ratio $w(x)$ which minimizes the Kullback-Leibler divergence between the reweighted source distribution $w(x)p_s(x)$ and the target distribution $p_t(x)$:

$$w^* = \underset{w}{\operatorname{argmin}} \operatorname{KL}\left(p_t(x), w(x)p_s(x)\right)$$

Optimization : Writing w as a parameterized function $w(x) = w_\theta(x)$, the optimization algorithm solved by KLIEP is written :

$$\theta^* = \underset{\theta \in \mathbb{R}^b}{\operatorname{argmax}} \sum_{x' \in \mathcal{T}_\mathcal{X}} \log\left(w_\theta(x)\right)$$

$$\text{sc} \sum_{x \in \mathcal{S}_\mathcal{X}} w_\theta(x) = 1 \text{ and } w_\theta(x) \geq 0$$

Kernel Mean Matching

Kernel Mean Matching (KMM)

The KMM algorithm looks for the density ratio $w(x)$ which minimizes the Maximum Mean Discrepancy (MMD) between the reweighted source distribution $w(x)p_s(x)$ and the target distribution $p_t(x)$:

$$w^* = \underset{w}{\operatorname{argmin}} \operatorname{MMD}\left(p_t(x), w(x)p_s(x)\right)$$

Optimization : Using a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, the optimization algorithm solved by KMM is written :

$$w^* = \underset{w \in \mathbb{R}^{m\times 1}; w \geq 0, w^T \mathbf{1} = m}{\operatorname{argmin}} \quad w^T K w - 2\kappa^T w$$

Where $K \in \mathbb{R}^{m\times m}$ and $\kappa \in \mathbb{R}^{m\times 1}$ such that : $K_{ij} = k(x_i, x_j)$ and $\kappa_i = \frac{m}{n} \sum_{x' \in \mathcal{T}_\mathcal{X}} k(x_i, x')$ with $x_i, x_j \in \mathcal{S}_\mathcal{X}$

Importance Weighting Classifier

The underlying idea of the Importance Weighting Classifier algorithm is to relate the density ratio $w(x)$ to the probability that x belongs to the source domain rather than the target domain. This probability is computed with a classifier $\phi : \mathcal{X} \to [0, 1]$ trained to discriminate between $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{T}_{\mathcal{X}}$:

$$\phi^* = \underset{\phi \in \Phi}{\operatorname{argmin}} \sum_{x \in \mathcal{S}_{\mathcal{X}}} \mathbb{1}\left(\phi(x) > \frac{1}{2}\right) + \sum_{x \in \mathcal{T}_{\mathcal{X}}} \mathbb{1}\left(\phi(x) < \frac{1}{2}\right)$$

$$w(x) = \frac{1}{\phi(x)} - 1$$

With Φ a set of classifiers.

Remark : The choice of the classifier set Φ drives the complexity of the weighting w (ex : linear models, neural networks, random forest...)

Importance Weighting Classifier

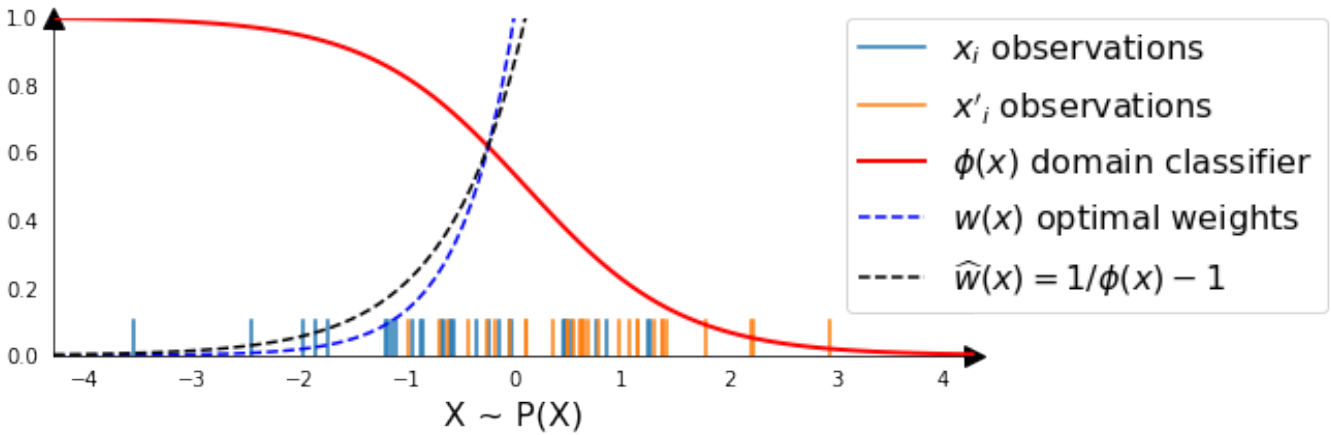


Figure – Here, a linear classifier is used to discriminate between the source (blue) and target (orange) samples.

Importance Weighting Classifier

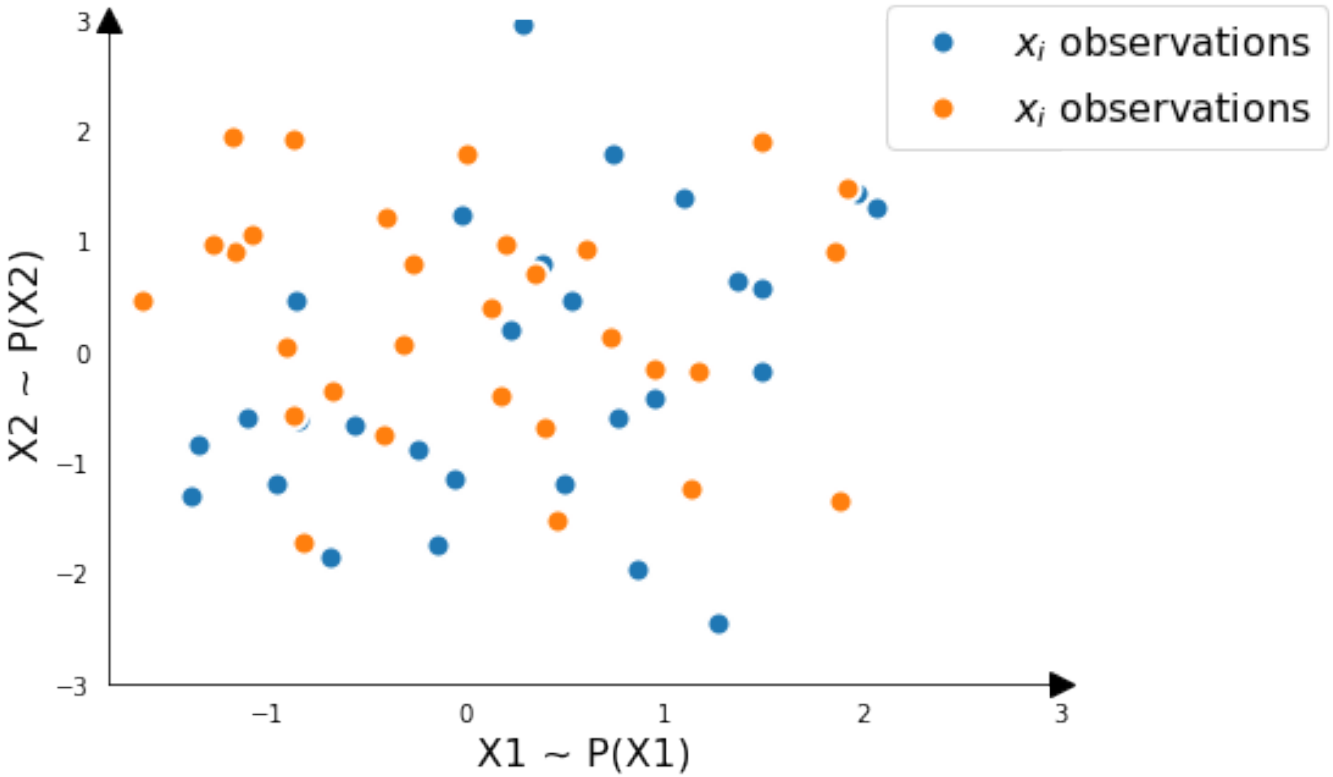


Figure – Here, a neural network classifier is used to discriminate between the source (blue) and target (orange) samples.

Importance Weighting Classifier

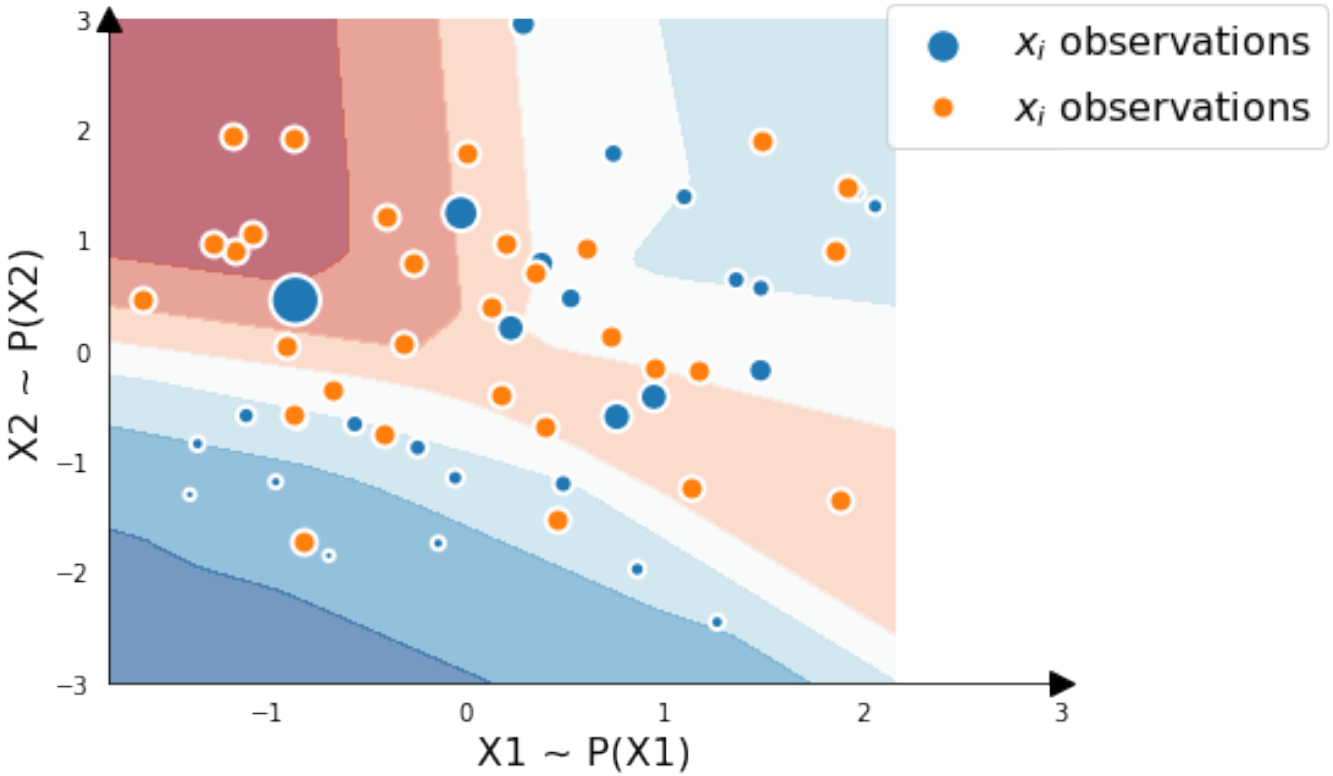


Figure – Here, a neural network classifier is used to discriminate between the source (blue) and target (orange) samples.

Exercise : Importance Weighting Classifier

Setup

Let's consider the source and target densities $p_s(x)$ and $p_t(x)$ such that $\text{supp}(p_t(x)) \subset \text{supp}(p_s(x))$. Let's define $\phi^*(x)$ as the best possible domain classifier with respect to the cross-entropy loss function :

$$\phi^* = \underset{\phi}{\text{argmax}}\, \mathbb{E}_{P_S(X)}[\log(\phi(X))] + \mathbb{E}_{P_T(X)}[\log(1 - \phi(X))]$$

Question : Show that, for any $x \in \mathcal{X}$:

$$w(x) = \frac{1}{\phi^*(x)} - 1$$

With $w(x) = \frac{p_t(x)}{p_s(x)}$

Exercise : Importance Weighting Classifier

Solution :

$$\begin{aligned}\phi^* &= \underset{\phi}{\text{argmax}}\, \mathbb{E}_{P_S(X)}[\log(\phi(X))] + \mathbb{E}_{P_T(X)}[\log(1-\phi(X))] \\ &= \underset{\phi}{\text{argmax}} \int_{x\in\mathcal{X}} p_s(x)\log(\phi(x))dx + \int_{x\in\mathcal{X}} p_t(x)\log(1-\phi(x))dx \\ &= \underset{\phi}{\text{argmax}} \int_{x\in\mathcal{X}} \left(p_s(x)\log(\phi(x)) + p_t(x)\log(1-\phi(x))\right)dx\end{aligned}$$

For any $x \in \mathcal{X}$, we can show that $\log(\phi(x)) + p_t(x)\log(1-\phi(x))$ is maximized for $\phi^*(x) = \frac{p_s(x)}{p_s(x)+p_t(x)}$. Then, we conclude that :

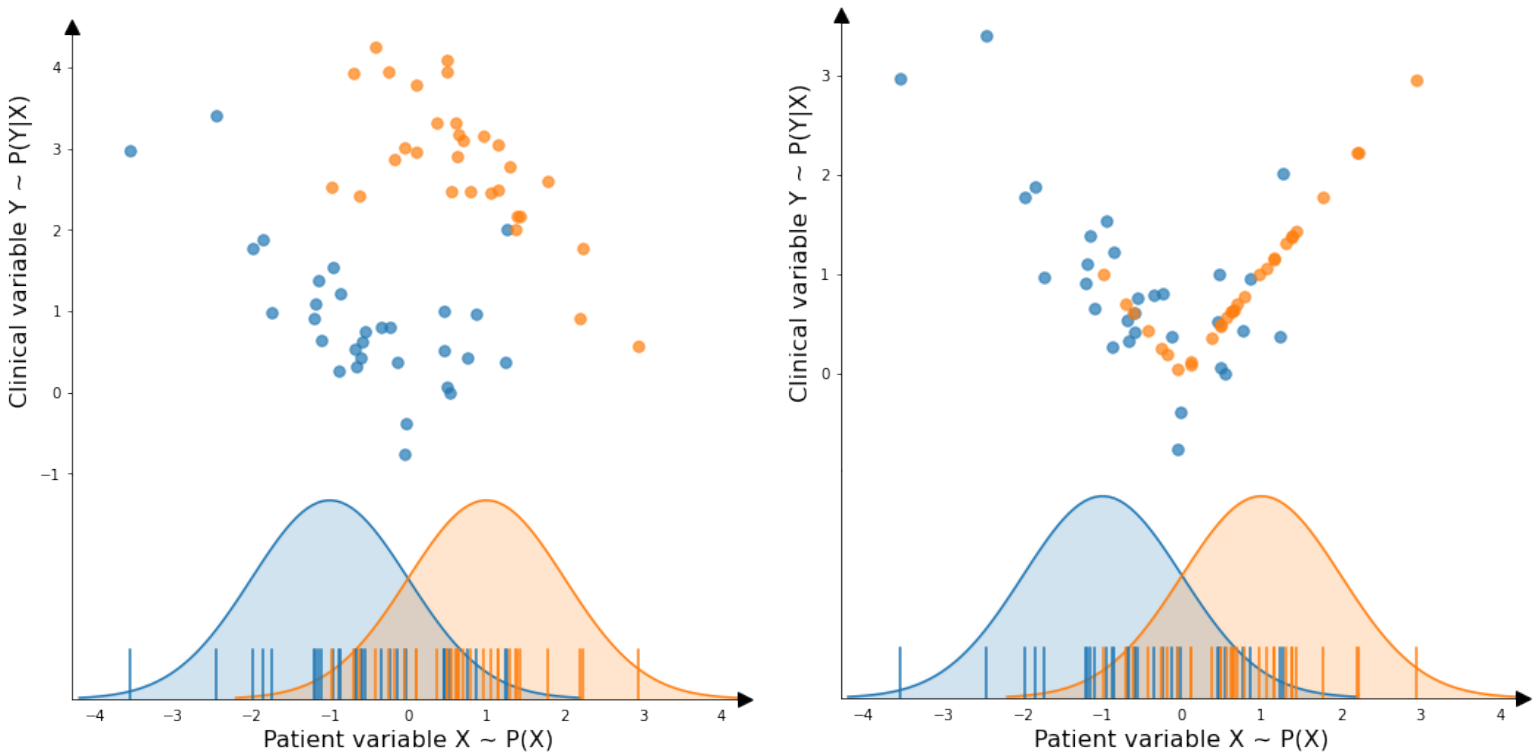
$$w(x) = \frac{1}{\phi^*(x)} - 1$$

What if no covariate-shift ?

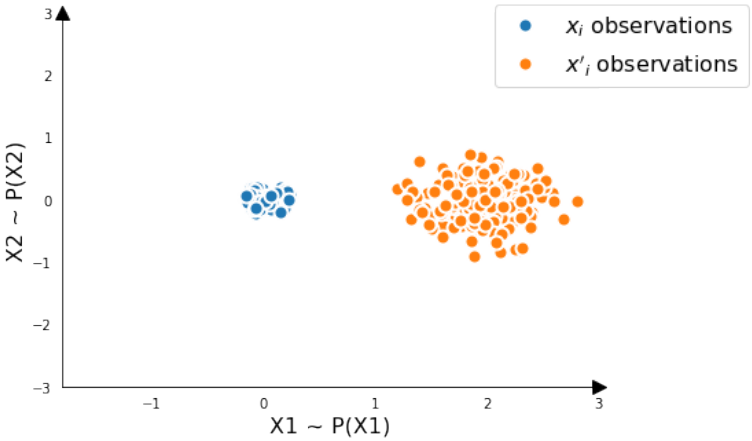
Definition 6 (Covariate Shift)

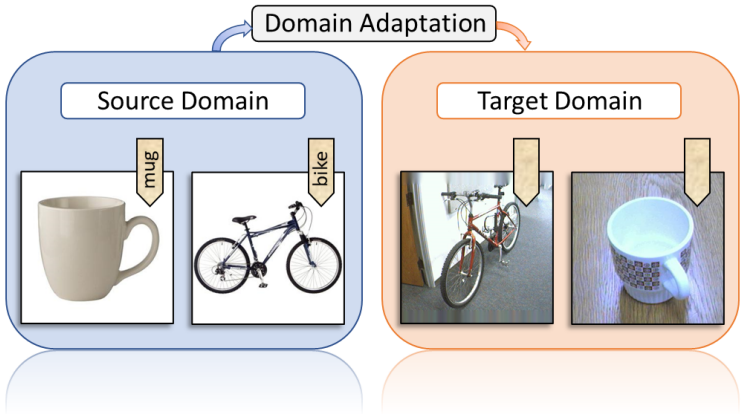
The source and target joint distributions $P_S(X, Y)$ and $P_T(X, Y)$ follows the **covariate-shift** assumption if, for any $x \in \mathcal{X}$:

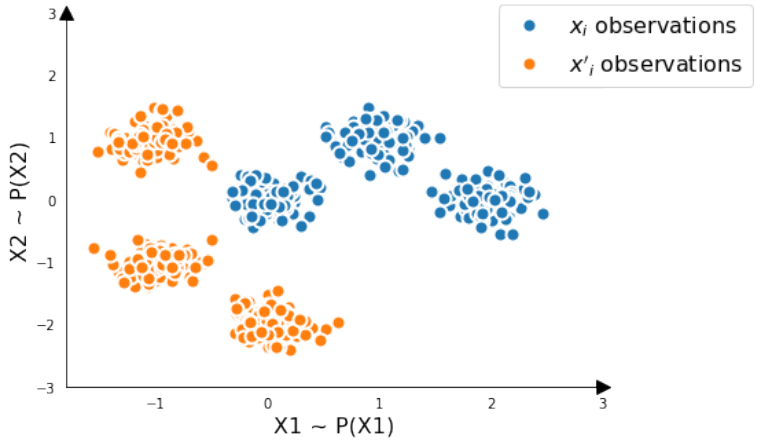
$$P_T(Y|X = x) = P_S(Y|X = x) \tag{8}$$

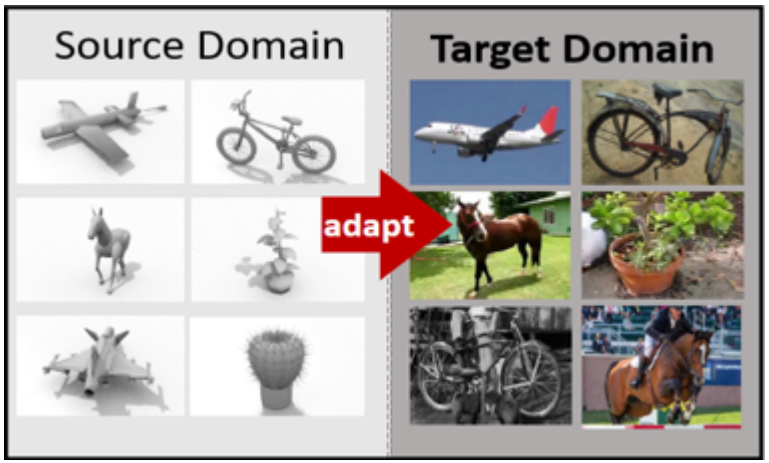


What if no common support ?

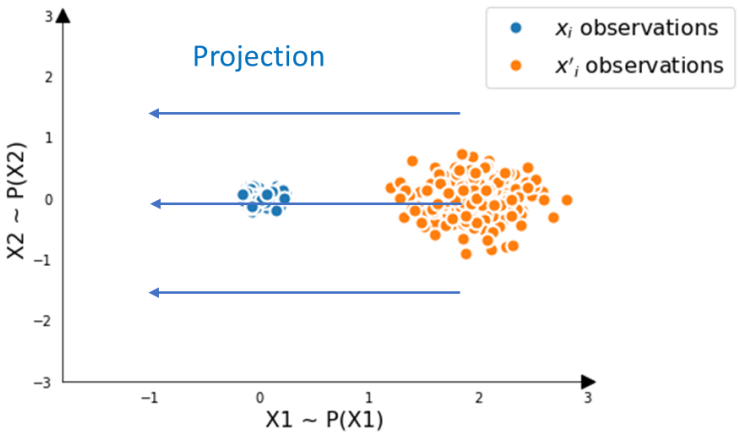




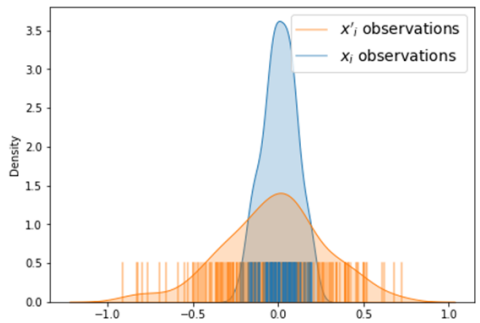


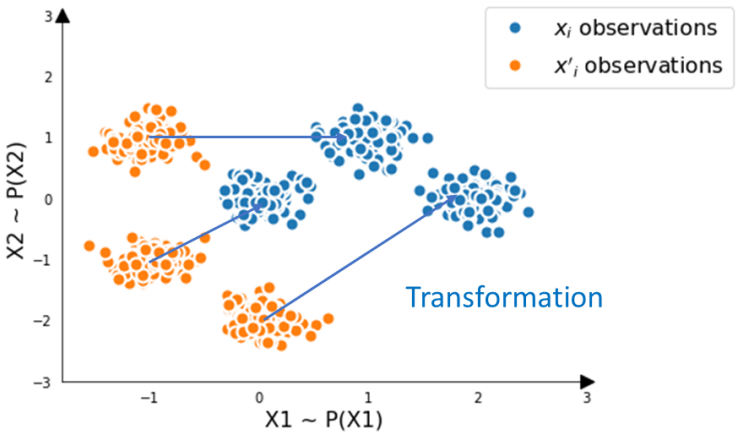


Opening : feature-based approach

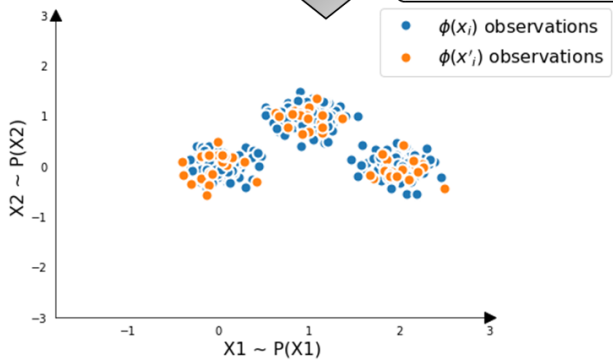


Feature Selection





Feature Transformation



Take Away

When should I use Importance Weighting ?

1. **Target labels are limited or unavailable :** You have a labeled source sample and unlabeled or sparsely labeled target sample.
2. **Source domain covers the target domain :** The support of the source distribution should include the support of the target distribution.
3. **Covariate-shift assumption holds :** The conditional distribution of labels given features is the same across source and target : $P_s(Y \mid X) = P_t(Y \mid X)$.
4. **Using a low-complexity or regularized model.** Or part of the source data may be irrelevant or noisy, containing outliers etc.

Take Away

Which Importance Weighting method should I use :

1. **Known parametric distribution :** Use Parametric Density Estimation to model the source and target densities directly.
2. **Distance-based approach :** If you have a meaningful distance or are fine with Euclidean, methods like KMM, KLIEP or NNW can be applied.
3. **Few samples ($<$ 10k) :** Kernel Mean Matching (KMM) or Nearest Neighbor Weighting (NNW) are suitable for small datasets.
4. **Moderate samples ($<$ 100k) :** KLIEP or NNW with approximation methods work well.
5. **Very large datasets :** Consider Neural Network domain classifier weighting.

Bibliography

Bickel, S., Brückner, M., and Scheffer, T. (2007).
Discriminative learning for differing training and test distributions.
In *Proceedings of the 24th international conference on Machine learning*, pages 81–88.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. J. (2007).
Correcting sample selection bias by unlabeled data.
In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press.

Loog, M. (2012).
Nearest neighbor-based importance weighting.
In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE.

Sugiyama, M., Nakajima, S., Kashima, H., Bünau, P. v., and Kawanabe, M. (2007).
Direct importance estimation with model selection and its application to covariate shift adaptation.
In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1433–1440, Red Hook, NY, USA. Curran Associates Inc.

Zhang, J., Ding, Z., Li, W., and Ogunbona, P. (2018).
Importance weighted adversarial nets for partial domain adaptation.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8156–8164.