Introduction Transfer Learning (2)

A. de Mathelin, M. Atiq

Michelin - Centre Borelli, ENS Paris-Saclay

December 2022

(Recap) UDA : Definition

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

Feature-based approach

Let Φ be a space of feature transformation : $\phi \in \Phi, \phi : \mathcal{X} \rightarrow \mathcal{X}$.
Feature-based approaches assume that there exist $\phi \in \Phi$ such that :

$$P_T(\phi(X), Y) = P_S(\phi(X), Y) \ \ \text{(symetric)} \tag{1}$$

or

$$P_T(\phi(X), Y) = P_S(X, Y) \ \ \text{(assymetric)} \tag{2}$$

UDA Feature-based

A UDA feature-based approach consist in solving the following optimization problems :

$$\widehat{\phi^*} = \underset{\phi \in \Phi}{\text{argmin}}\, D(\phi(\mathcal{S}_\mathcal{X}), \phi(\mathcal{T}_\mathcal{X}))$$

$$\widehat{h^*} = \underset{h \in \mathcal{H}}{\text{argmin}} \sum_{(x_i, y_i) \in \mathcal{S}} \ell(h(\widehat{\phi^*}(x_i)), y_i)$$

With D a discrepancy measure between distributions.

Feature-based Examples

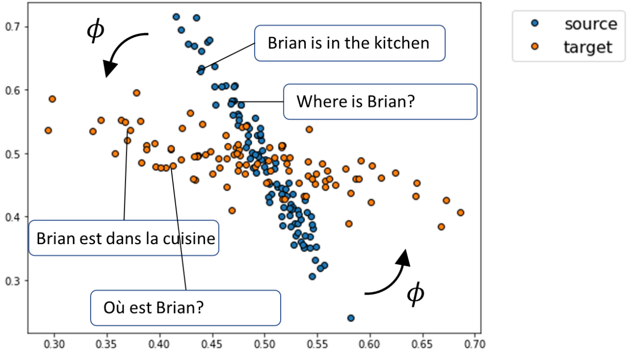


Figure – **Rotation** In this example, the space of feature transformation is $\Phi = \{x \to Mx^T; M \in \mathbb{R}^{p \times p}, M^T M = \mathrm{Id}_p\}$ and is used to match a dataset of english sentences to a dataset of french sentences.

Feature-based Examples

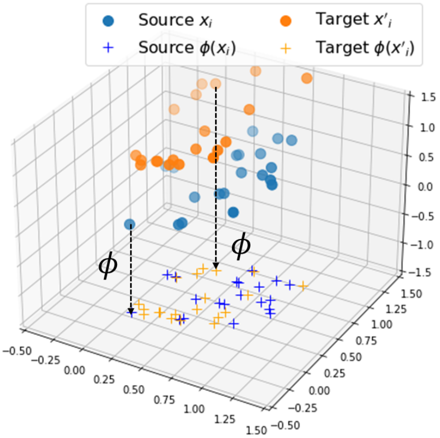


Figure – **Projection** In this example, the space of feature transformation is $\Phi = \{x \to Px^T; P \in \text{Proj}(\mathbb{R}^{p \times p})\}$

Feature-based Examples

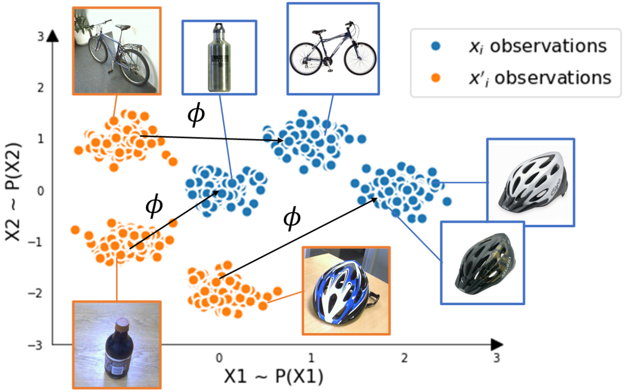


Figure – **Continuous Transformation** In this example, the space of feature transformation is $\Phi = \mathcal{C}_0(\mathcal{X})$ and is used to match a dataset of pictures from Amazon to a dataset of webcam pictures.

Feature-based Examples

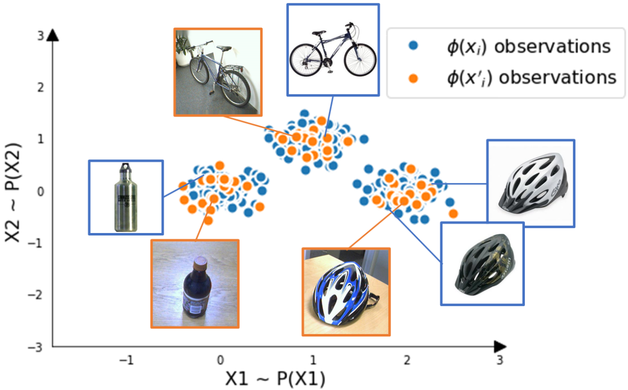


Figure – **Continuous Transformation** In this example, the space of feature transformation is $\Phi = \mathcal{C}_0(\mathcal{X})$ and is used to match a dataset of pictures from Amazon to a dataset of webcam pictures.

Feature-based approach : Subspace Alignment

Subspace Alignment (SA)

The purpose of Subspace Alignment [Fernando et al., 2013] is to find a linear transformation of the source PCA eigenvectors which match as close as possible the target PCA eigenvectors :

$$\widehat{\phi_T^*} = x \to xW_T^d$$
$$\widehat{\phi_S^*} = x \to xW_S^d M^*$$
$$M^* = \underset{M \in \mathbb{R}^{d\times d}}{\operatorname{argmin}} ||W_S^d M - W_T^d||_2^2$$

Where W_T^d and W_S^d are respectively the matrixes of the d first eigenvectors of the target and source PCA.

Subspace Alignment Example

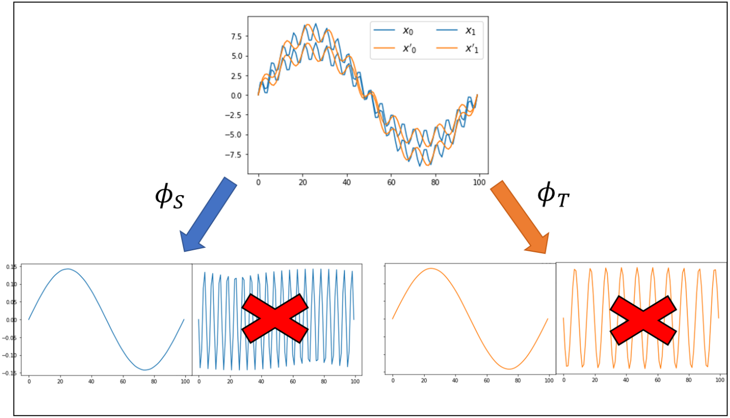


Figure – **Subspace Alignment** In this example, the source dataset is composed of signals : $t \to 5X_1 \sin(2\pi t) + X_2 \sin(40\pi t)$ and the target dataset of signals : $t \to 5X_1 \sin(2\pi t) + X_2 \sin(20\pi t)$ $(X_1, X_2 \sim \mathcal{U}([0, 2]))$

Correlation Alignment (CORAL)

The purpose of Correlation Alignment [Sun et al., 2016] is to find a linear transformation of the source data which covariance matrix match as cloas as possible the target data covariance matrix :

$$\widehat{\phi_S^*} = x \to xA^*$$

$$A^* = \underset{A \in \mathbb{R}^{p \times p}}{\operatorname{argmin}} ||A^T C_S A - C_T||_2^2$$

Where C_T and C_S are respectively the covariance matrixes of the target and source dataset.

Correlation Alignment Example

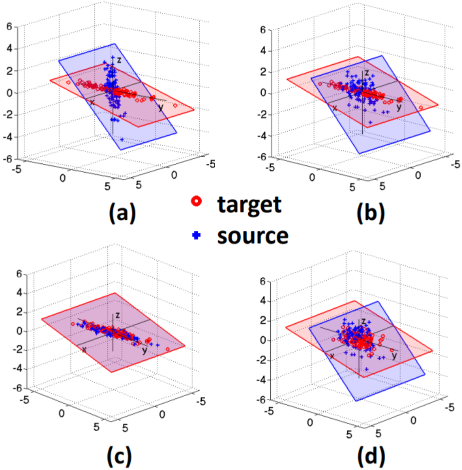


Figure – **Correlation Alignment** (source [Sun et al., 2016]) (a) : initial situation, (b) : source data are "whitened", (c) : source data are "re-colored" with the target covariance.

Feature-based approach : Optimal Transport

Optimal Transport for Domain Adaptation (OTDA)

The purpose of OTDA [Courty et al., 2016] is to find the optimal transportation from the source data to the target data :

$$\widehat{\phi_S^*} = x \to \sum_{x' \in \mathcal{T}} \gamma^*(x, x') x'$$

$$\gamma^* = \underset{\gamma \in \Gamma}{\text{argmin}} \sum_{x \in \mathcal{S}} \sum_{x' \in \mathcal{T}} \gamma(x, x') ||x - x'||_2^2$$

Where, for any $\gamma \in \Gamma$, $\gamma : \mathcal{S} \times \mathcal{T} \to [0, 1]$ and :

$$\sum_{x \in \mathcal{S}} \gamma(x, x') = 1 \ \forall x' \in \mathcal{T}$$

$$\sum_{x' \in \mathcal{T}} \gamma(x, x') = 1 \ \forall x \in \mathcal{S}$$

OTDA Example

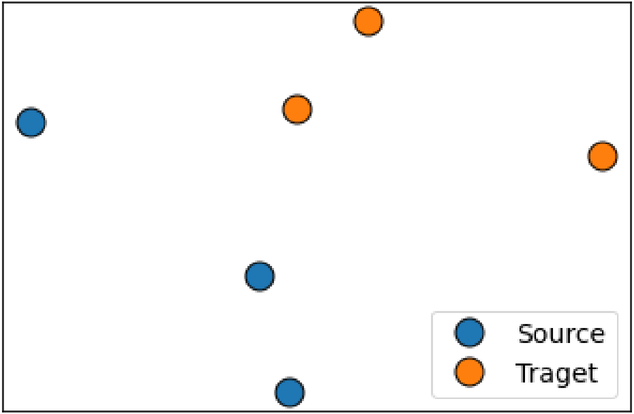


Figure – **OTDA** : In this example, we are looking at the optimal pairing between three source and target data points. The optimal pairing minimizes the sum of distances between paired data points.

OTDA Example

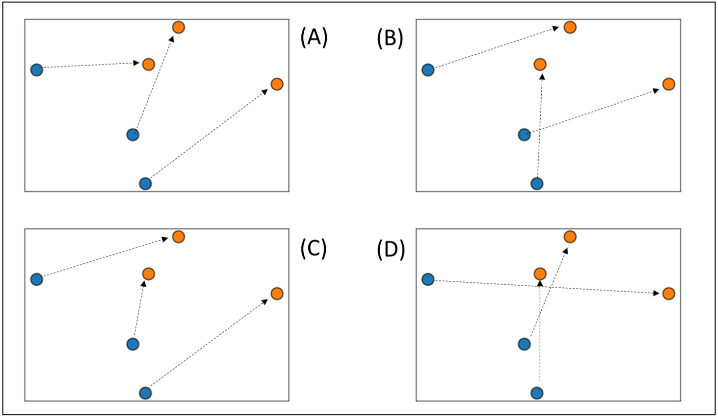


Figure – **OTDA** : In this example, we are looking at the optimal pairing between three source and target data points. The optimal pairing minimizes the sum of distances between paired data points.

OTDA Example

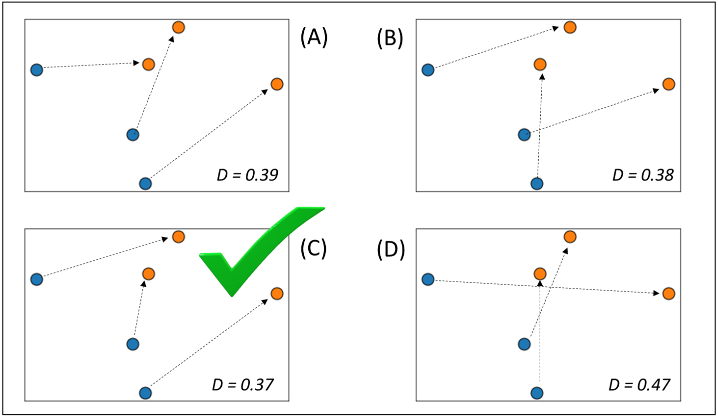


Figure – **OTDA** : The optimal pairing is the pairing (C). Finding the optimal pairing for large datasets is often intractable, we are then looking at approximated solution.

Outline

1. Feature-based Transfer Learning
2. Parameter-based Transfer Learning
3. Conclusion

Regularization Transfer LR

The purpose of Regularization Transfer LR [Chelba et al., 2007] is to find the target linear regression coefficients that fit target data while being "close" to the source coefficients :

$$\widehat{\beta^*_S} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} ||X_S \beta^T - Y_S||_2^2$$

$$\widehat{\beta^*_T} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} ||X_T \beta^T - Y_T||_2^2 + \lambda ||\beta - \beta_S||_2^2$$

With X_S, Y_S and X_T, Y_T the source and target input data and labels.

Exercise : Transfer Learning with Linear Regression

Let's consider $X_T \in \mathbb{R}^{n\times p}$, $Y_T = \mathbb{R}^{n\times 1}$ and $\beta_S \in \mathbb{R}^p$. Find a close-form solution to the problem :

$$\widehat{\beta_T^*} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} ||X_T\beta^T - Y_T||_2^2 + \lambda||\beta - \beta_S||_2^2$$

Hint : Write $||.||_2^2$ as a scalar product $\langle ., . \rangle$ and find the vector $D(\beta)$ such that :

$$G(\beta + h) - G(\beta) = \langle D(\beta), h\rangle + o(h)$$

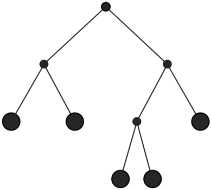
With $G(\beta) = ||X_T\beta^T - Y_T||_2^2 + \lambda||\beta - \beta_S||_2^2$ and $h \in \mathbb{R}^p$

The solution β_T^* verifies $D(\beta_T^*) = 0$

(Reminder) Decision Trees and Random Forests

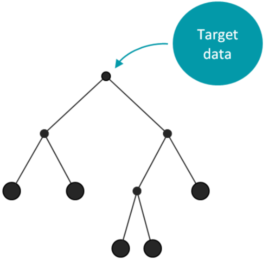
[Segev et al., 2017]

Structure Expansion Reduction (SER)



[Segev et al., 2017]

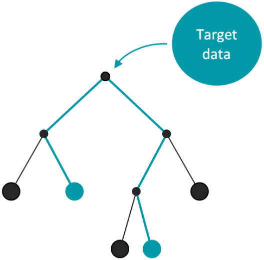
Structure Expansion Reduction (SER)



Supervised Transfer Learning on Decision Tree

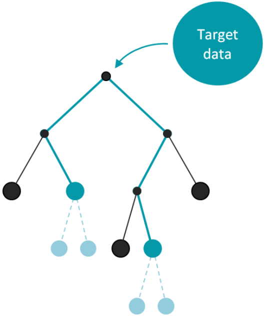
[Segev et al., 2017]

Structure Expansion Reduction (SER)



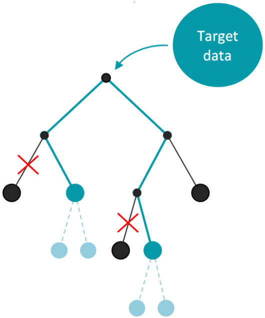
[Segev et al., 2017]

Structure Expansion Reduction (SER)



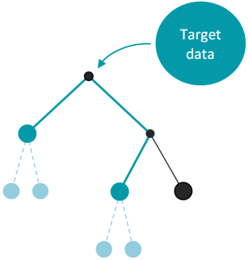
[Segev et al., 2017]

Structure Expansion Reduction (SER)



[Segev et al., 2017]

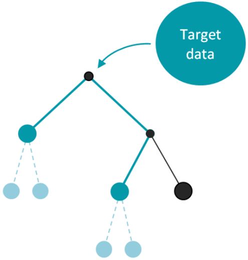
Structure Expansion Reduction (SER)

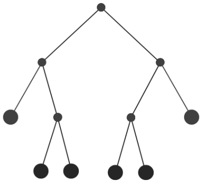


Supervised Transfer Learning on Decision Tree

[Segev et al., 2017]

Structure Expansion Reduction (SER) Structure Transfer (STRUT)

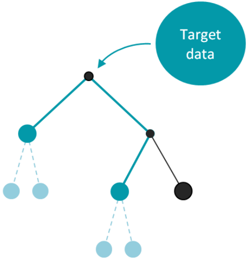




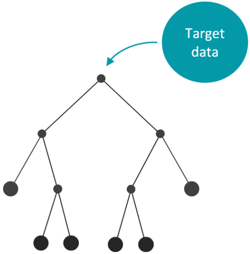
Supervised Transfer Learning on Decision Tree

[Segev et al., 2017]

Structure Expansion Reduction (SER)



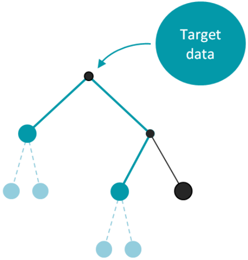
Structure Transfer (STRUT)



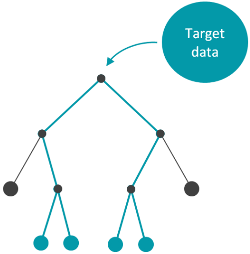
Supervised Transfer Learning on Decision Tree

[Segev et al., 2017]

Structure Expansion Reduction (SER)



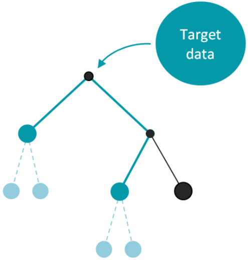
Structure Transfer (STRUT)



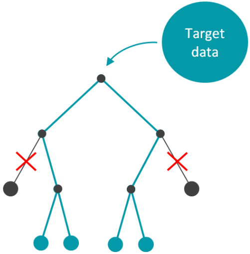
Supervised Transfer Learning on Decision Tree

[Segev et al., 2017]

Structure Expansion Reduction (SER)



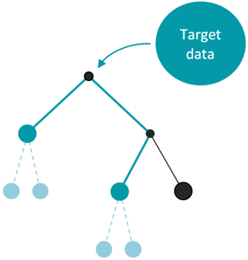
Structure Transfer (STRUT)



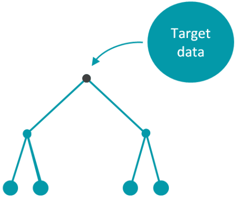
Supervised Transfer Learning on Decision Tree

[Segev et al., 2017]

Structure Expansion Reduction (SER)



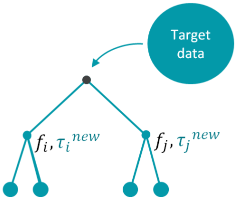
Structure Transfer (STRUT)



[Segev et al., 2017]

Structure Expansion Reduction (SER)

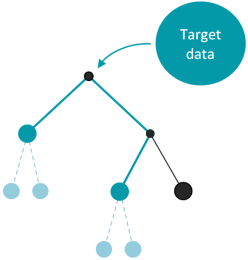
Structure Transfer (STRUT)



Supervised Transfer Learning on Decision Tree

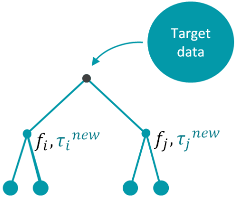
[Segev et al., 2017]

Structure Expansion Reduction (SER)



Partition refinement/simplification

Structure Transfer (STRUT)

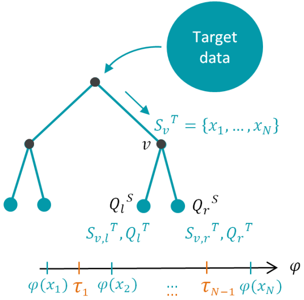


Local Drifts / Squeezes and Stretches

STRUT divergence gain optimization

Try to keep split left/right target distributions as close as possible to source ones by updating the threshold.

STRUT node update



Q_l^S, Q_r^S : class proportions of source data in children w.r.t. the **original** split.

$S_{v,l}^T$, $S_{v,r}^T$: subsets of S_v^T that fall in the children nodes of v.

$Q_l^T(\tau)$, $Q_r^T(\tau)$: class proportions of target data in children w.r.t. the **new** split.

Goal : Maximize DG while being in a local maximum of Information Gain (IG) (here IG = Gini gain).

DG optimization

$$\tau_m = \underset{\tau \in \mathrm{T}_v}{\operatorname{argmax}} \left(DG\left(Q_l^S, Q_r^S, Q_l^T(\tau), Q_r^T(\tau) \right) \right)$$

$$\text{s.t. } IG(\tau_{m-1}) < IG(\tau_m) \text{ and } IG(\tau_m) > IG(\tau_{m+1})$$

Divergence Gain :

$$DG(\tau) = 1 - \frac{|S_{v,l}^T(\tau)|}{|S_v^T|} J(Q_l^S, Q_l^T(\tau)) - \frac{|S_{v,r}^T(\tau)|}{|S_v^T|} J(Q_r^S, Q_r^T(\tau))$$

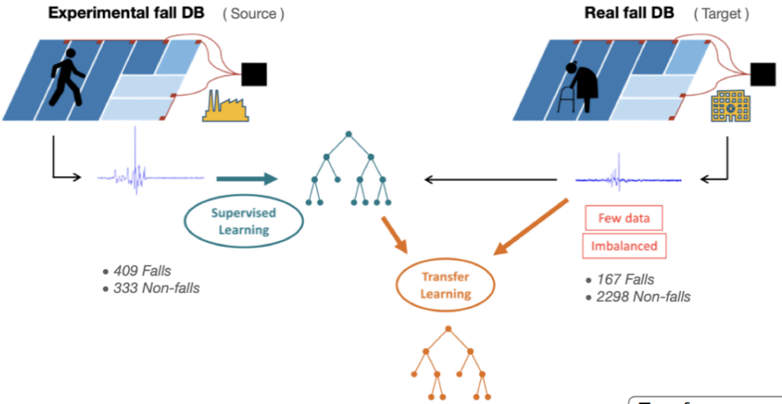
Jensen-Shannon divergence :

$$J(P, Q) = \frac{1}{2}\left(KL(P||M) + KL(Q||M)\right), \text{ with } M = \frac{1}{2}(P + Q)$$

Kullback-Leibler divergence :

$$KL(P||Q) = \sum_{k=1}^{K} P_k \ln\left(\frac{P_k}{Q_k}\right)$$

Transfer learning with Class Imbalance (Fall detection)



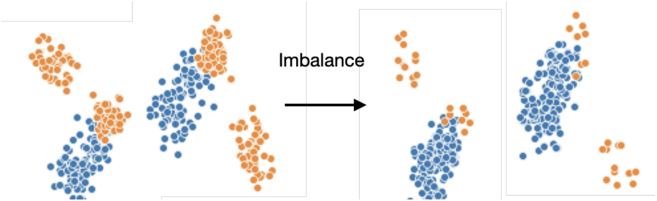
Source domain : $\mathcal{D}_S = \{\mathcal{X}_S, P^S(X_S)\}$
Target domain : $\mathcal{D}_T = \{\mathcal{X}_T, P^T(X_T)\}$
Source task : $\mathcal{T}_S = \{\mathcal{Y}_S, P^S(Y_S/X_S)\}$
Target task : $\mathcal{T}_T = \{\mathcal{Y}_T, P^T(Y_T/X_T)\}$

$\mathcal{X}_S = \mathcal{X}_T, \mathcal{Y}_S = \mathcal{Y}_T$
$P^S(X_S) \neq P^T(X_T)$
$P^S(Y_S/X_S) \neq P^T(Y_T/X_T)$

Transfer context :
Homogeneous
Inductive
Model-based

Transfer Learning with Class Imbalance

Class	Src	Targ	Ser	Strut
Maj class	10.9	6.3	13.8	4.8
Min class	10.7	**5.0**	**7.0**	**2.2**



SER and *STRUT* are not suited for target class imbalance.

Target shift

Conditional distribution for a given label y stays the same but the overall proportion of each label changes between source and target. $\forall y \in \mathcal{Y}$,

- $P_T(Y = y) \neq P_S(Y = y)$
- $P_T(X|Y = y) = P_S(X|Y = y)$

Comparably to **covariate shift**, the **density ratio** can be expressed depending on Y only ($w(X, Y) = w(Y)$).
We can deduce $P_T(Y|X)$ from $P_S(Y|X)$ and $w(Y)$.

$$
\begin{aligned}
P_T(X, Y) \quad &= P_T(X|Y)P_T(Y) = w(Y)P_S(Y)P_S(X|Y) = w(Y)P_S(X|Y) \\
P_T(Y|X) \quad &= \frac{w(Y)P_S(X, Y)}{P_T(X)} = \frac{w(Y)P_S(X, Y)}{\sum_{k=1}^{K} w(y_k)P_S(X, Y = y_k)} = \frac{w(Y)P_S(Y|X)P_S(X)}{\sum_{k=1}^{K} w(y_k)P_S(Y = y_k|X)P_S(X)} \\
&= \frac{w(Y)P_S(Y|X)}{\sum_{k=1}^{K} w(y_k)P_S(Y = y_k|X)}
\end{aligned}
$$

Transfer Learning with Class Imbalance

- What is the "simplest" target class imbalance situation ?
  - How to quantify the risk of pruning relevant leaves ?

Transfer Learning with Class Imbalance

- What is the "simplest" target class imbalance situation ?
  - How to quantify the risk of pruning relevant leaves ?

Homogeneous class imbalance :

$$p^T(x|y) = p^S(x|y) \tag{3}$$

$$p^T(y = k|x) = \lambda_k \frac{p^S(y = k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y = j|x)}, \text{with} \quad \lambda_k = \frac{P^T(y = k)}{P^S(y = k)} \tag{4}$$

Transfer Learning with Class Imbalance

- What is the "simplest" target class imbalance situation ?
  - How to quantify the risk of pruning relevant leaves ?

Homogeneous class imbalance :

$$p^T(x|y) = p^S(x|y) \tag{3}$$

$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}, \text{with} \quad \lambda_k = \frac{P^T(y=k)}{P^S(y=k)} \tag{4}$$

Pruning risk :

Goal : *Estimate the risk of loosing a source leaf l of minority class k_m with n_m target data.*
Minority class leaves that should stay unchanged :

$$\forall k \neq k_m, p^T(y=k_m|x \in l) > p^T(y=k|x \in l)$$

Transfer Learning with Class Imbalance

- What is the "simplest" target class imbalance situation ?
  - How to quantify the risk of pruning relevant leaves ?

Homogeneous class imbalance :

$$p^T(x|y) = p^S(x|y) \tag{3}$$

$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}, \text{with} \quad \lambda_k = \frac{P^T(y=k)}{P^S(y=k)} \tag{4}$$

Pruning risk :

Goal : *Estimate the risk of loosing a source leaf l of minority class k_m with n_m target data.*
Minority class leaves that should stay unchanged :

$$\forall k \neq k_m, p^T(y=k_m|x \in l) > p^T(y=k|x \in l)$$

$$\boxed{\overset{(4)}{\Longrightarrow} \quad \lambda_{k_m} p^S(y=k_m|x \in l) > \lambda_k p^S(y=k|x \in l)}$$

Transfer Learning with Class Imbalance

- What is the "simplest" target class imbalance situation ?
  - How to quantify the risk of pruning relevant leaves ?

Homogeneous class imbalance :

$$p^T(x|y) = p^S(x|y) \tag{3}$$

$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}, \text{with} \quad \lambda_k = \frac{P^T(y=k)}{P^S(y=k)} \tag{4}$$

Pruning risk :

Goal : *Estimate the risk of loosing a source leaf l of minority class k_m with n_m target data.*
Minority class leaves that should stay unchanged :

$$\forall k \neq k_m, p^T(y=k_m|x \in l) > p^T(y=k|x \in l)$$

$$\boxed{\overset{(4)}{\Longrightarrow} \quad \lambda_{k_m} p^S(y=k_m|x \in l) > \lambda_k p^S(y=k|x \in l)}$$

Risk of leaf l being unreached by n_m minority class target data :

$$PRR_{n_m}(l) \overset{def}{=} p^T(\text{No data among } n_m \text{ of class } k_m \text{ class reaching } l) = p^T(x \notin l|y=k_m)^{n_m}$$

Transfer Learning with Class Imbalance

- What is the "simplest" target class imbalance situation ?
  - How to quantify the risk of pruning relevant leaves ?

Homogeneous class imbalance :

$$p^T(x|y) = p^S(x|y) \tag{3}$$

$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}, \text{with} \quad \lambda_k = \frac{P^T(y=k)}{P^S(y=k)} \tag{4}$$

Pruning risk :

Goal : *Estimate the risk of loosing a source leaf l of minority class k_m with n_m target data.*
Minority class leaves that should stay unchanged :

$$\forall k \neq k_m, p^T(y=k_m|x \in l) > p^T(y=k|x \in l)$$

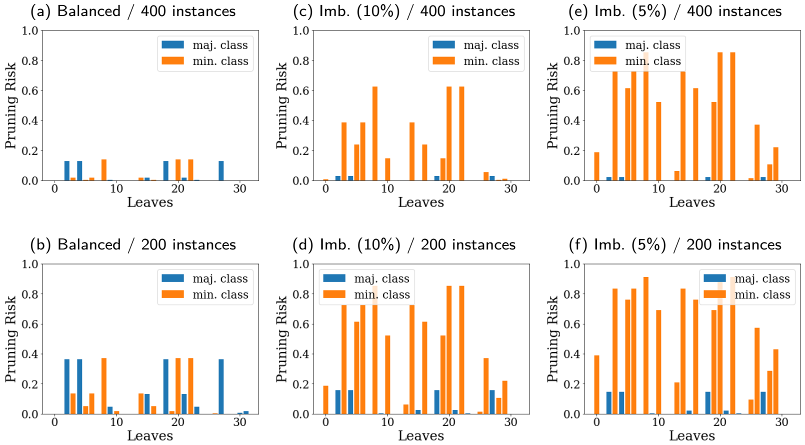
$$\boxed{\overset{(4)}{\Longrightarrow} \quad \lambda_{k_m} p^S(y=k_m|x \in l) > \lambda_k p^S(y=k|x \in l)}$$

Risk of leaf l being unreached by n_m minority class target data :

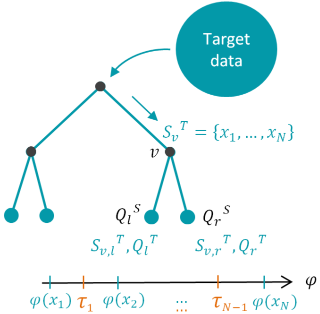
$$PRR_{n_m}(l) \overset{def}{=} p^T(\text{No data among } n_m \text{ of class } k_m \text{ class reaching } l) = p^T(x \notin l|y=k_m)^{n_m}$$

$$\boxed{\overset{(3)}{\Longrightarrow} \quad PRR_{n_m}(l) = p^S(x \notin l|y=k_m)^{n_m}}$$

The pruning risk increases rapidly with class imbalance and data rarity.



STRUT node update



Q_l^S, Q_r^S : class proportions of source data in children w.r.t. the **original** split.

$S_{v,l}^T$, $S_{v,r}^T$: subsets of S_v^T that fall in the children nodes of v.

$Q_l^T(\tau)$, $Q_r^T(\tau)$: class proportions of target data in children w.r.t. the **new** split.

DG optimization

$$\tau_m = \operatorname*{argmax}_{\tau \in \mathrm{T}_v} \left(DG \left(Q_l^S, Q_r^S, Q_l^T(\tau), Q_r^T(\tau) \right) \right)$$

s.t. $IG(\tau_{m-1}) < IG(\tau_m)$ and $IG(\tau_m) > IG(\tau_{m+1})$

STRUT(λ)

Use of equation (2) :

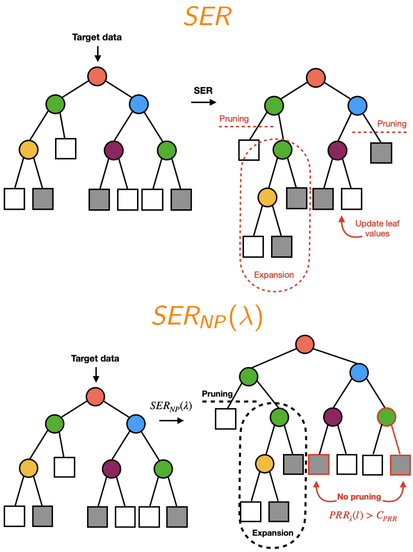
$$p^T(y = k|x) = \lambda_k \frac{p^S(y = k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y = j|x)}$$

to change the source class proportions in DG :

$$Q_{l,k}^{S'} = \lambda_k \frac{Q_{l,k}^S}{\sum_k \lambda_j Q_{l,j}^S} \qquad Q_{r,k}^{S'} = \lambda_k \frac{Q_{r,k}^S}{\sum_j \lambda_j Q_{r,j}^S}$$

STRUT(λ) can be seen as a generalization (original *STRUT* $\Leftrightarrow \lambda \simeq 1$)

SER generalization for Class Imbalance [Minvielle et al., 2019]



$PRR_{n_m}(l) = p^S(x \notin l|k_m)^{n_m}$: (1)

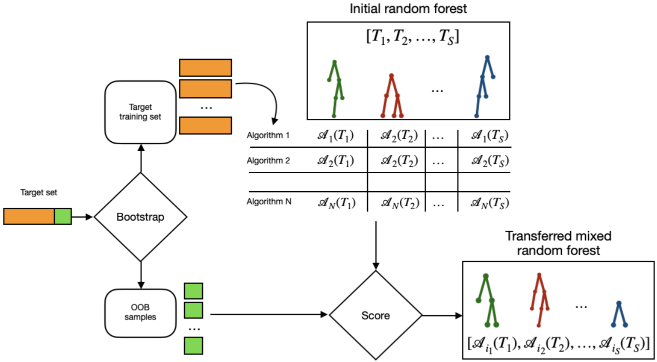
$\lambda_{k_m} p^S(k_m|x \in l) > \lambda_k p^S(k|x \in l)$: (2)

Adaptation for class imbalance :

- Expansion step unchanged
- Flag each leaf l following Eq. (2) and with $PRR_{n_m}(l) > C_{PRR}$
- Avoid any pruning involving leaf l

Same pruning limitation strategy on STRUT : $STRUT_{NP}(\lambda)$

Selective Transferred Random Forest (STRF)



Returns : A mixed random forest using several transfer algorithms and a selection ratio for each algorithm.

Conclusion

Bibliography

Chelba, C., Silva, J., and Acero, A. (2007).
Soft indexing of speech content for search in spoken documents.
Comput. Speech Lang., 21(3) :458–478.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016).
Optimal transport for domain adaptation.
IEEE transactions on pattern analysis and machine intelligence, 39(9) :1853–1865.

Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. (2013).
Unsupervised visual domain adaptation using subspace alignment.
In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967.

Minvielle, L., Atiq, M., Peignier, S., and Mougeot, M. (2019).
Transfer learning on decision tree with class imbalance.
In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1003–1010.

Segev, N., Harel, M., Mannor, S., Crammer, K., and El-Yaniv, R. (2017).
Learn on source, refine on target : A model transfer learning framework with random forests.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(9) :1811–1824.

Sun, B., Feng, J., and Saenko, K. (2016).
Return of frustratingly easy domain adaptation.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.