Introduction Transfer Learning

A. de Mathelin [1]
M. Atiq [2]

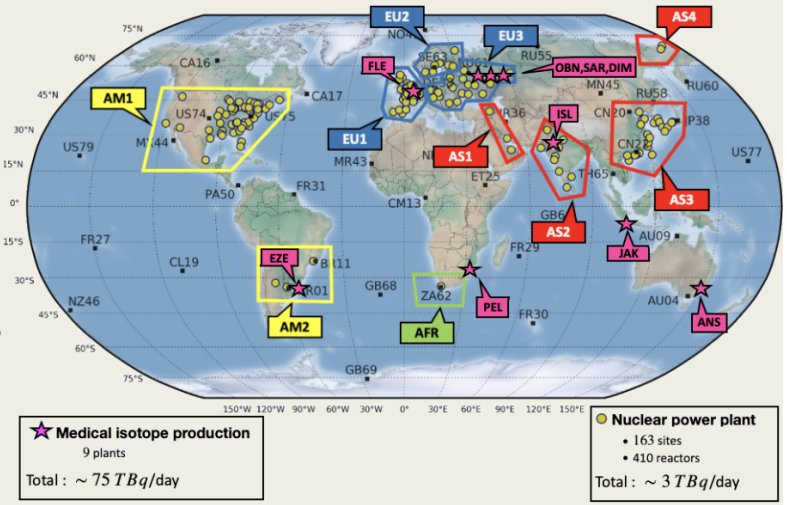
Sloan Kettering Institute - CEA

December 2025

1. antoine.demat@gmail.com
2. atiq.mounir@gmail.com

My current work

- Precise characterisation of Xenon-133 atmospheric background is crucial to detect anomalies.
- New approach to represent this background on simulated data:
  - Simulations of Xenon-133 concentrations on period 2020-2023
  - Emitters gathered in 19 distinct sources
  - Graph model representing the IMS network
- Graphs allow to cluster the IMS stations into regional areas based on source distribution.
- Method to estimate source contributions from total concentrations on each station.
- First step to develop new anomaly detection methods.



My current work

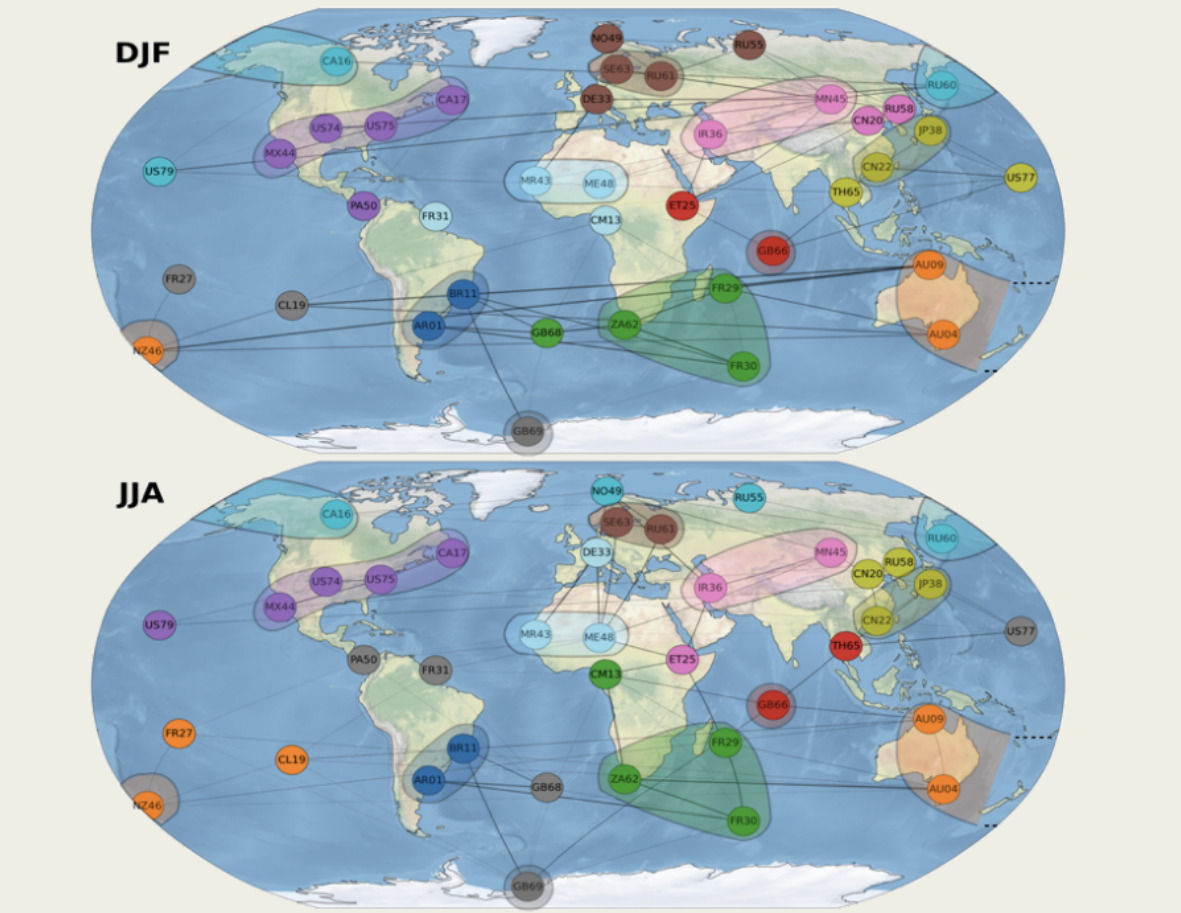


Figure 2. Spectral clustering of IMS stations by season : December-February (DJF) and June-August (JJA). Each colour represents a cluster.

Figure – Seasonal station clustering

My current work

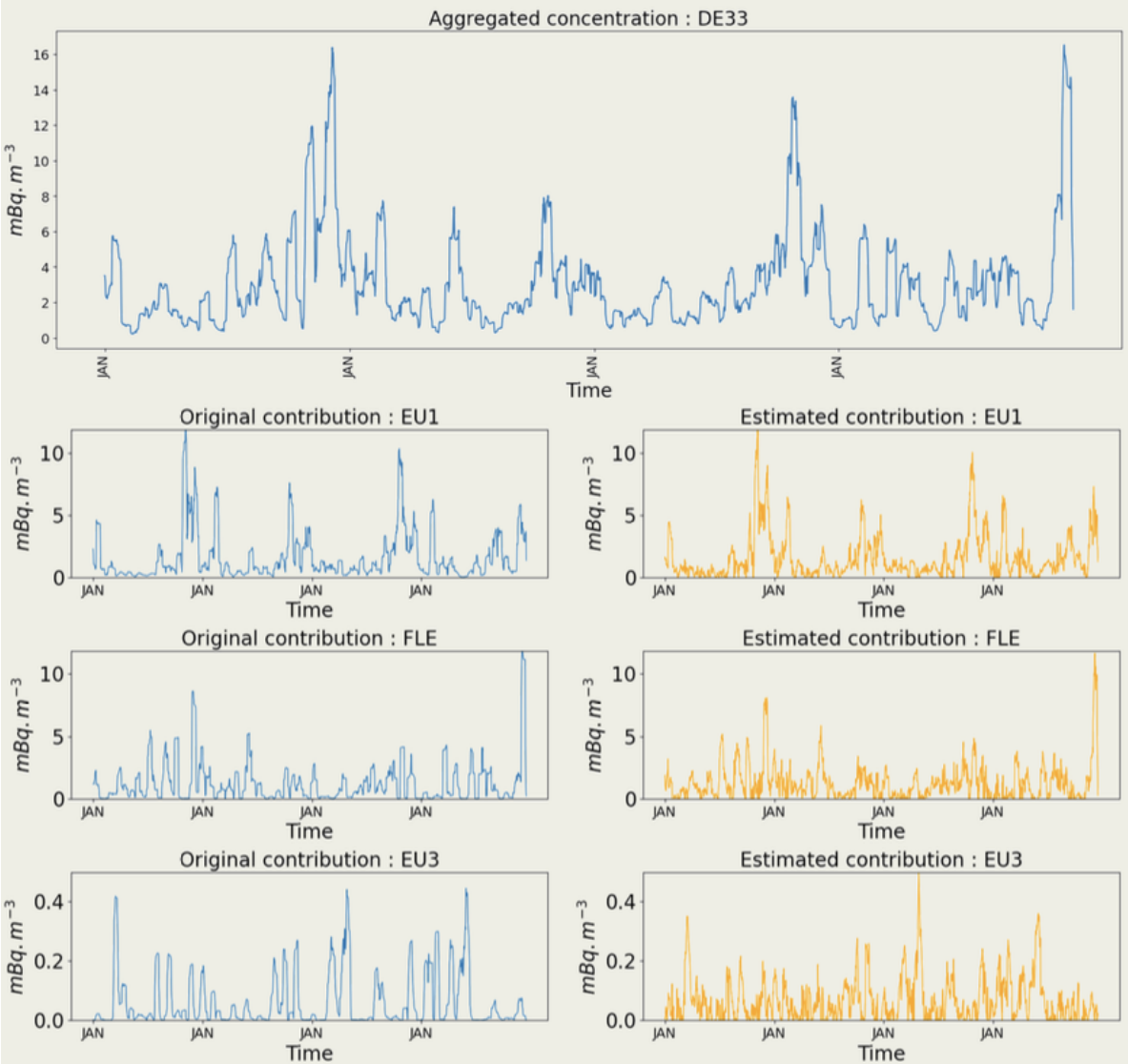


Figure – Xenon emissions source separation

Outline

Feature-based approach : augmenting the feature space

Feature augmentation

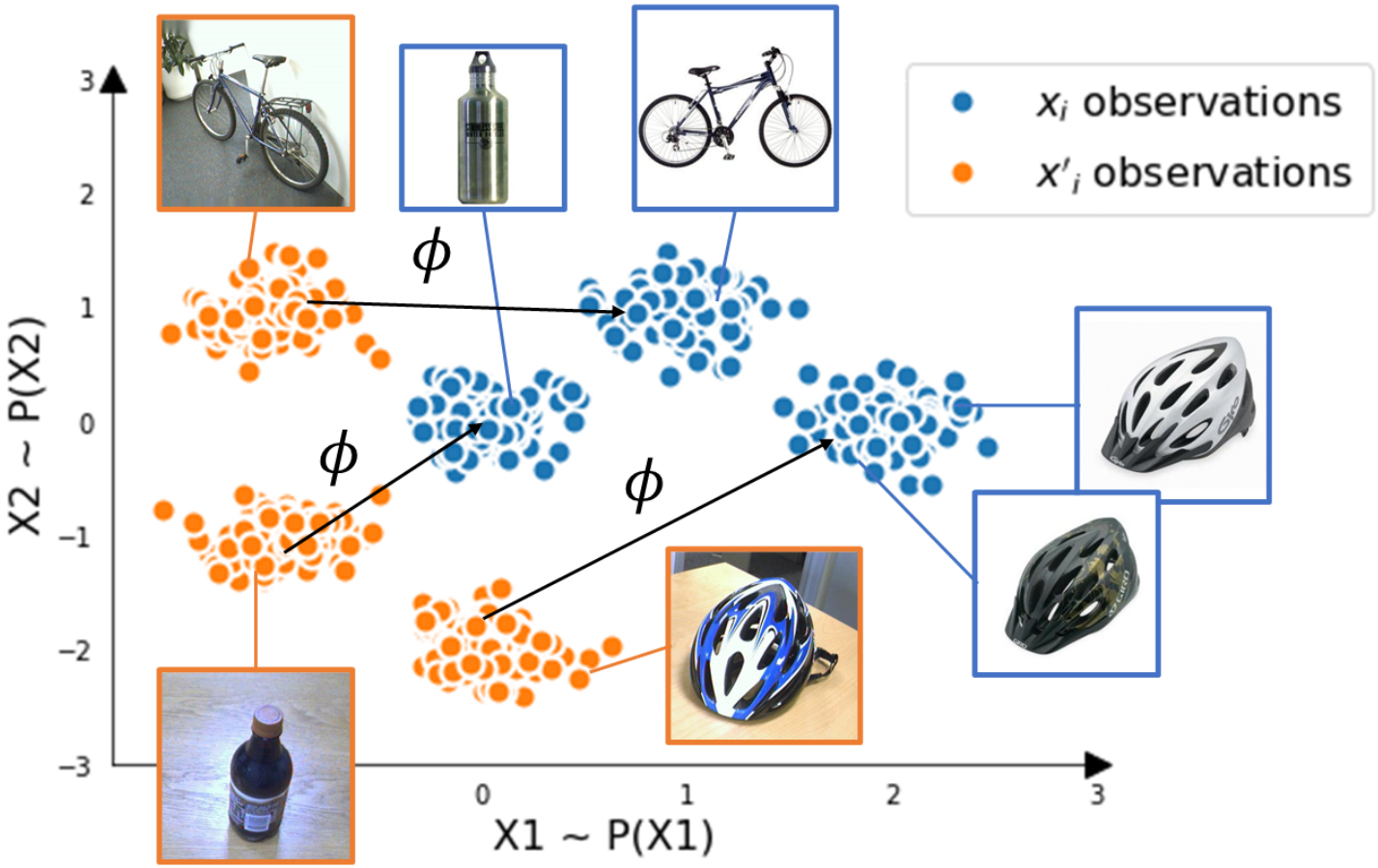
Feature augmentation [Daumé III, 2007] is to duplicate Source and Target features, firstly considering them as different variables and secondly as common variables.

- X_S becomes $(X_S, 0, X_s)$
- X_T becomes $(0, X_T, X_T)$

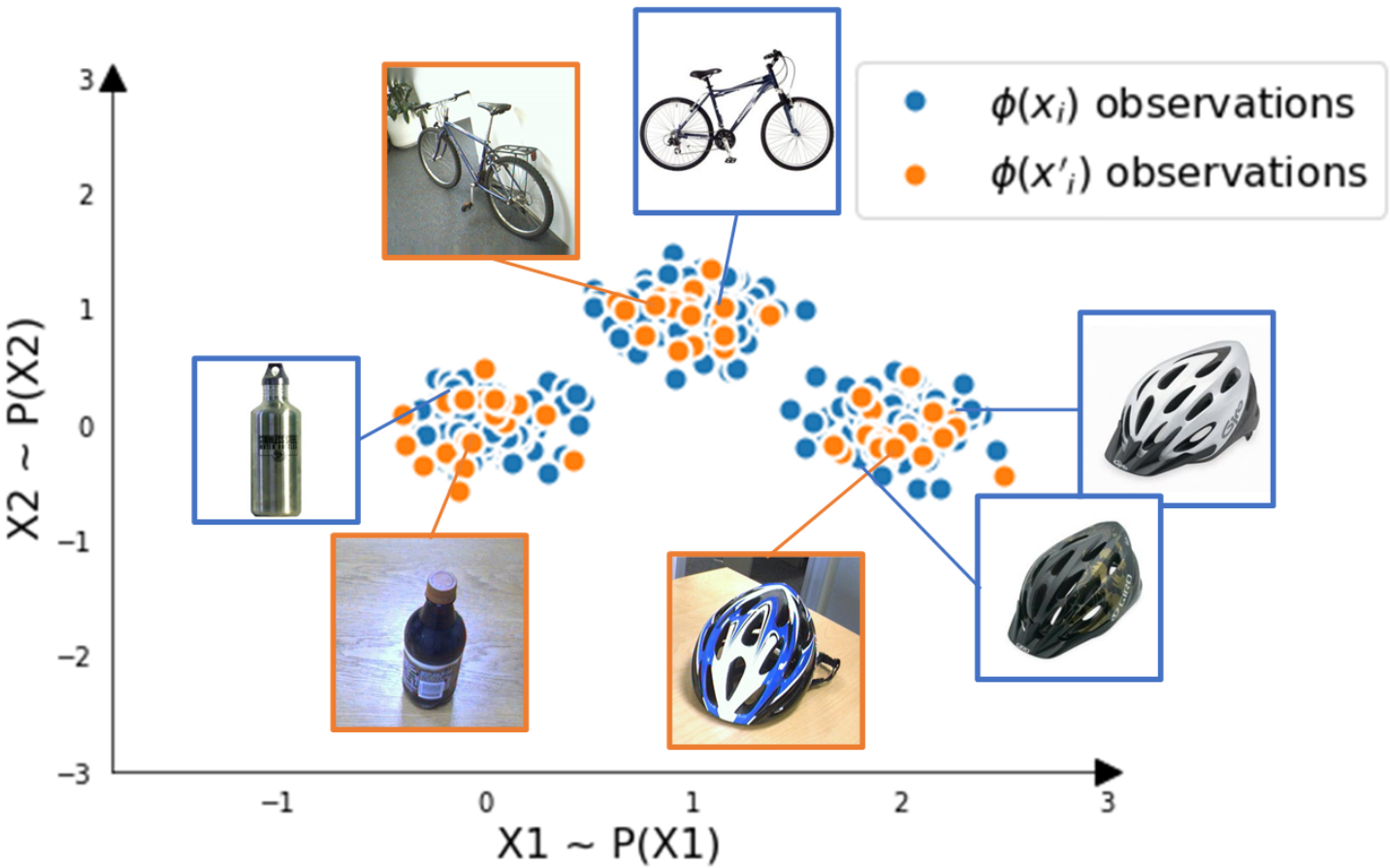
This pre-processing produces 3 times more features corresponding to 3 blocks :

- Source specific features
- Target specific features
- General common features

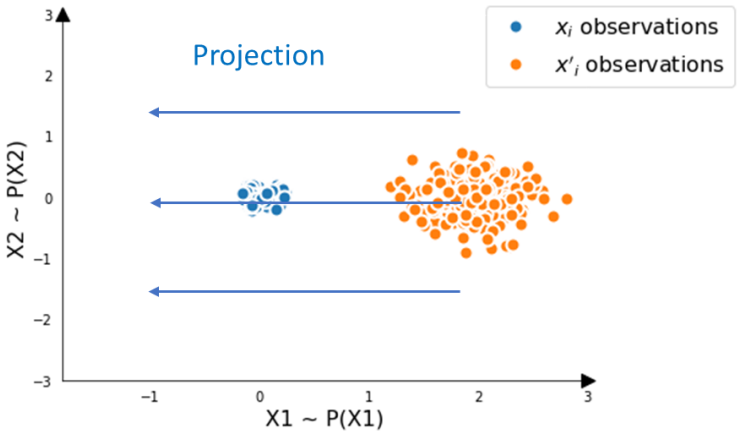
Opening : feature-based alignment approach



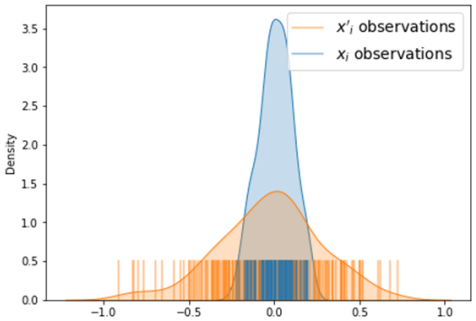
Opening : feature-based alignment approach

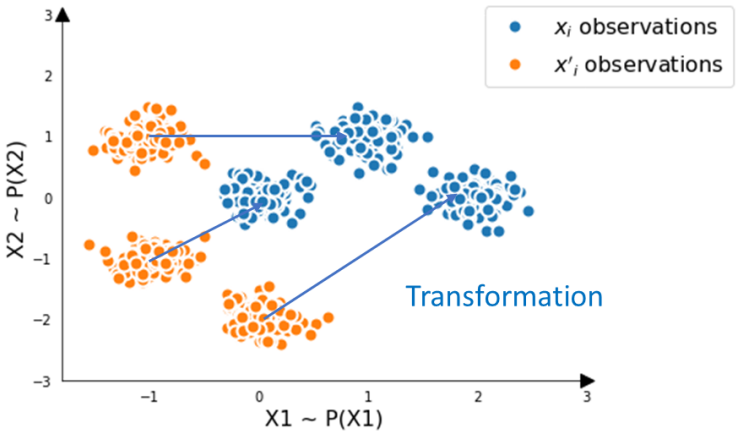


Opening : feature-based alignment approach

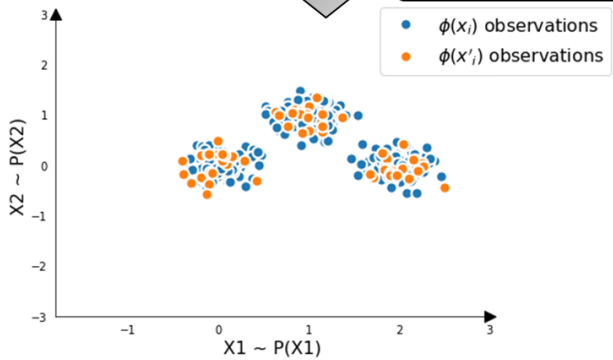


Feature Selection





Feature Transformation



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 close as possible the target data covariance matrix :

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

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

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

Correlation Alignment Example

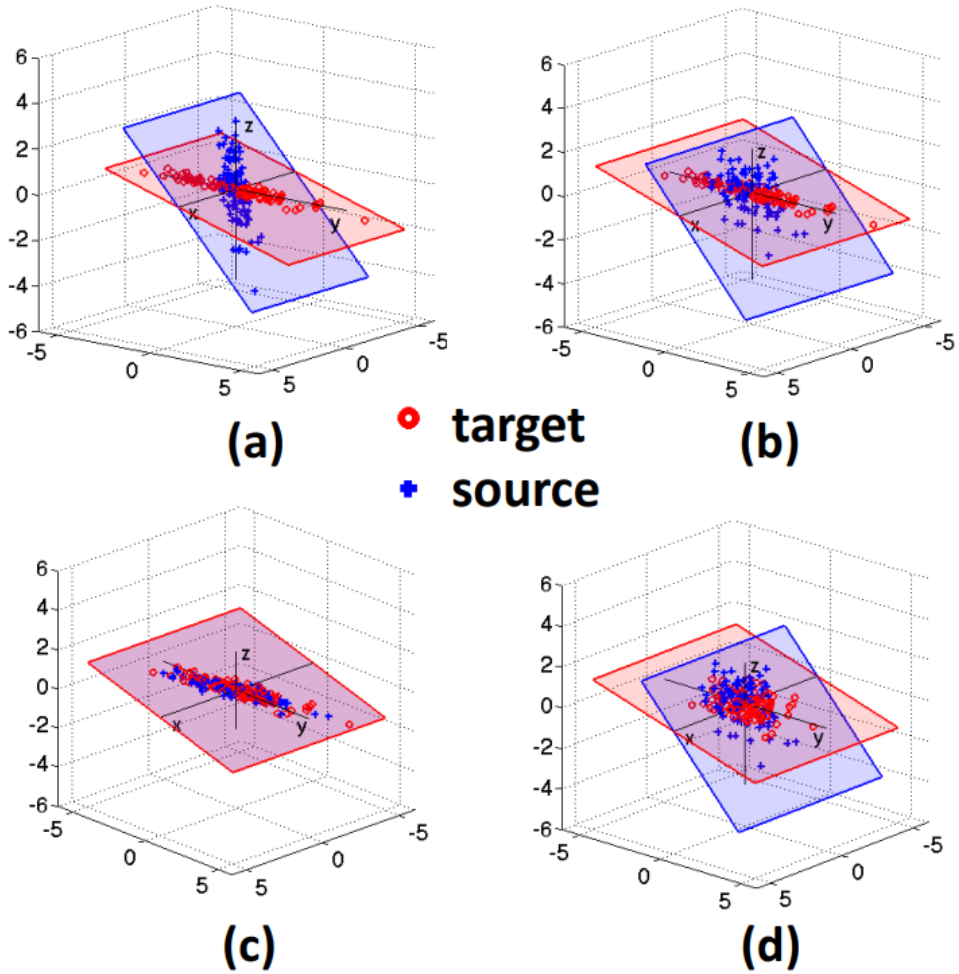
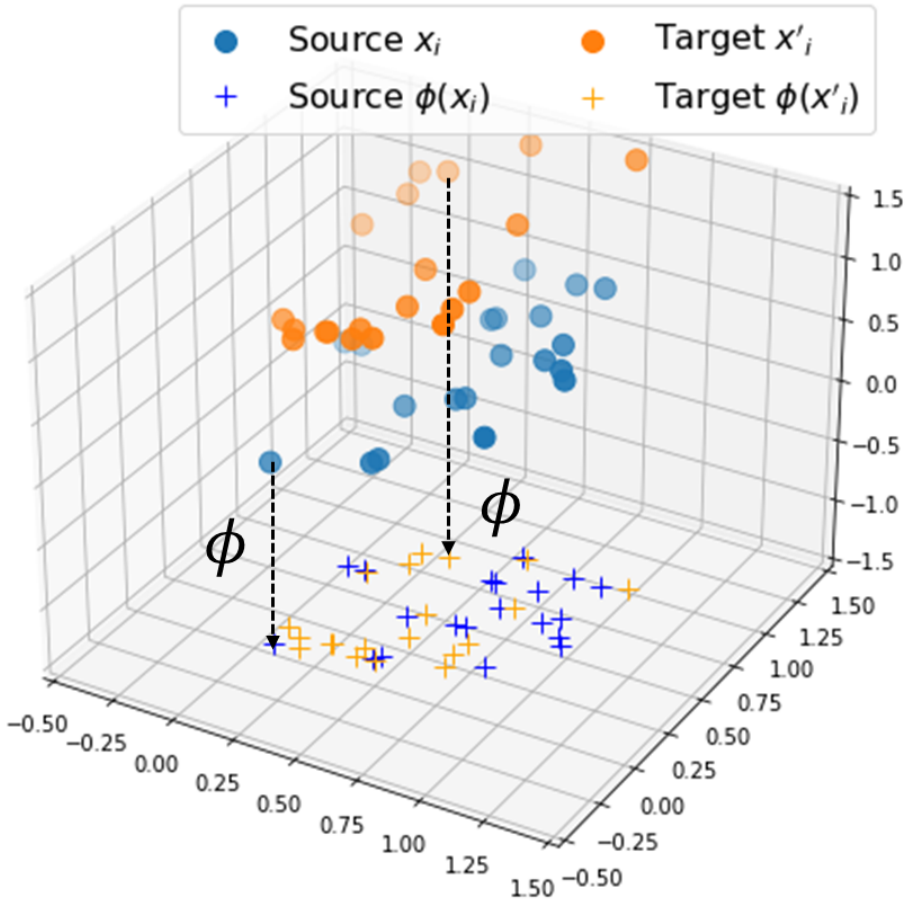


Figure – **Correlation Alignment** (source [Sun et al., 2016]) (a) : initial situation, (b) : source data are "whitened", (c) : source data are "re-colored" with the target covariance. (d) Whitening both would result in no alignment.

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 :

$$\widehat{\phi_T} : x \to xV_T^d$$

$$\widehat{\phi_S} : x \to xV_S^d M^*$$

$$M^* = \underset{M \in \mathbb{R}^{d \times d}}{\text{argmin}} \, ||V_S^d M - V_T^d||_F^2$$

Where V_T^d and V_S^d are respectively the matrices of the d first eigenvectors of the target and source PCA.

PCA subspace alignment

Advantages :

- Do not need any label
- Easy to compute
- Works with any model

How to choose d ?

- Bounds criteria based on concentration inequalities gives a d_{max} for a given confidence [Fernando et al., 2013] .
- "Hardness" of the domain adaptation can be assessed by a divergence measure $d_{H\Delta H}(P_{X_S}, P_{X_T})$ [Ben-David et al., 2010]

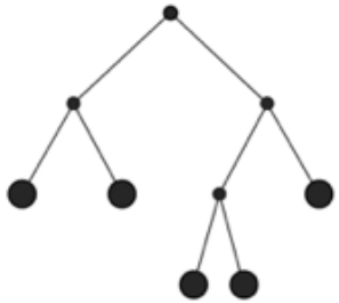
In the **supervised** setting, we can choose d empirically.

Outline

Transfer learning approaches : model-based

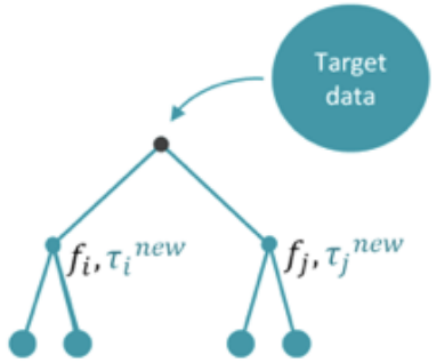
Pre-trained **Source** model, updated by **Target** data.

Initial source model



Updated by target data

Constraint on parameter Deviation



We want to control the "dissimilarity" between Source / Target model parameters.

Regularization Transfer LR

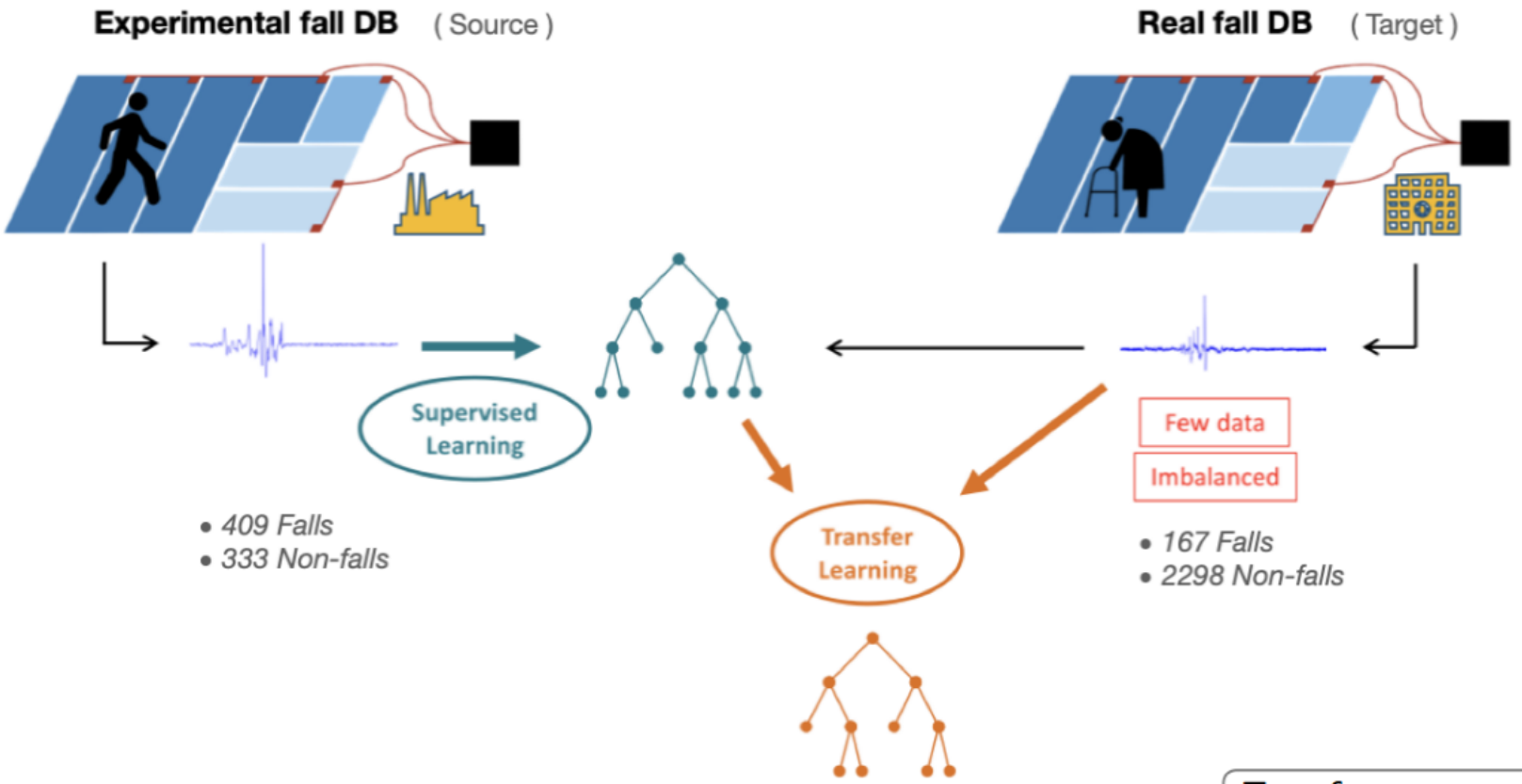
Regularization Transfer LR [Chelba et al., 2007] tries to find the target linear regression coefficients that fit target data while being "close" to the source coefficients :

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

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

Same idea for linear classifiers.

TP2 : Fall detection data



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 $:\mathcal{T}_T = \{\mathcal{Y}_T, P^T(Y_T/X_T)\}$

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

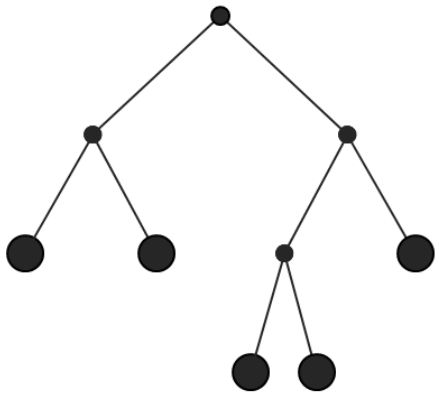
Transfer context :
Homogeneous
Inductive
Model-based

A Data Set for Fall Detection with Smart Floor Sensors. [Truong et al., 2023]

Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

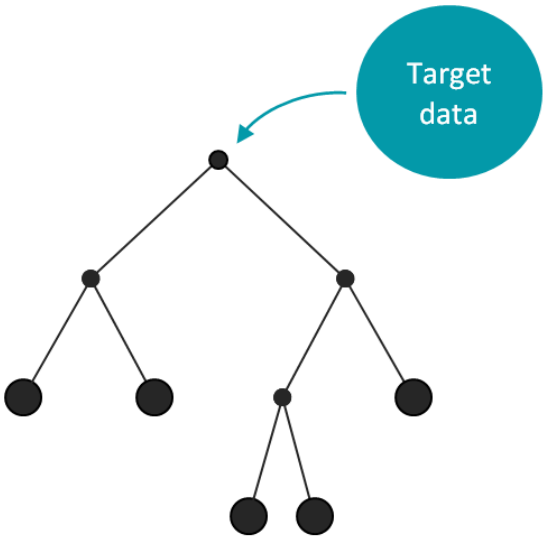
Structure Expansion Reduction (SER)



Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

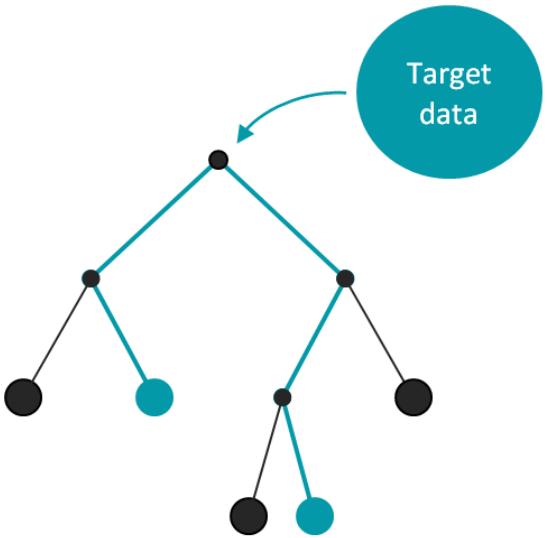
Structure Expansion Reduction (SER)



Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

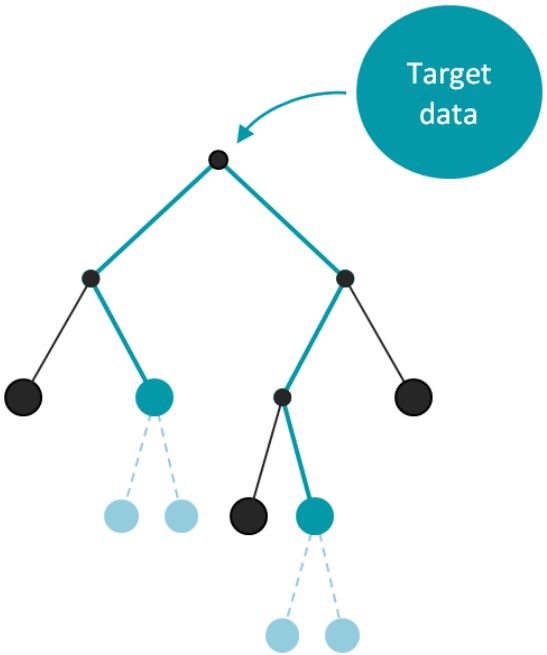
Structure Expansion Reduction (SER)



Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

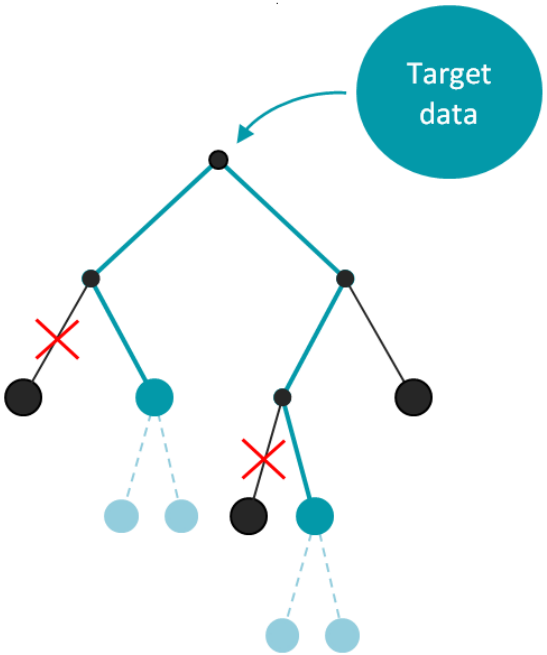
Structure Expansion Reduction (SER)



Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

Structure Expansion Reduction (SER)



Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

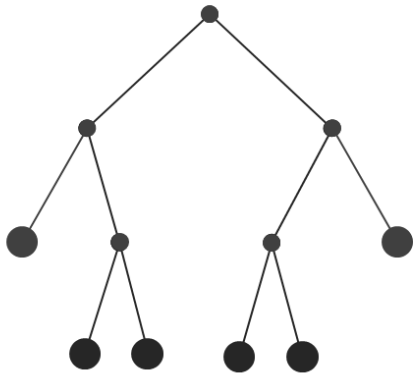
Structure Expansion Reduction (SER)

Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

Structure Expansion Reduction (SER)

Structure Transfer (STRUT)

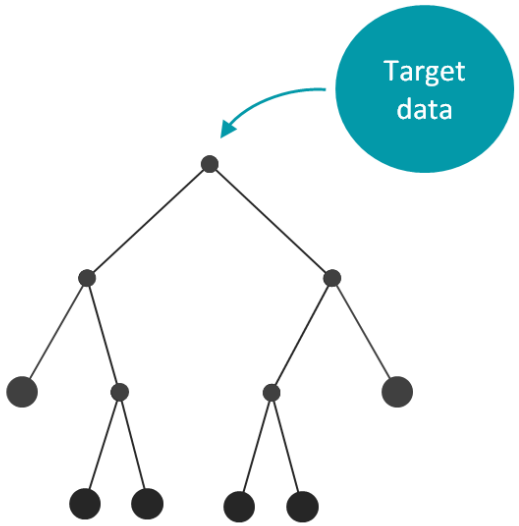


Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

Structure Expansion Reduction (SER)

Structure Transfer (STRUT)

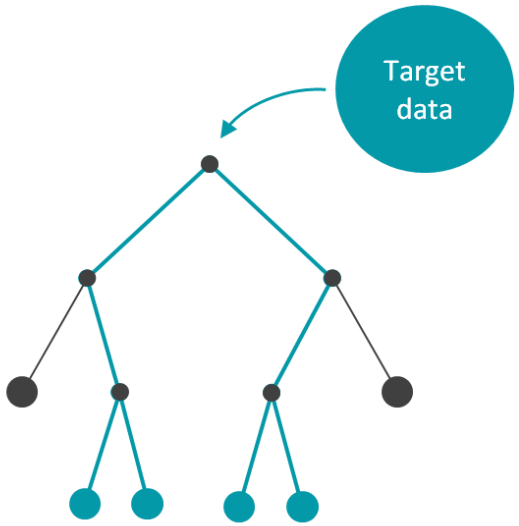


Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

Structure Expansion Reduction (SER)

Structure Transfer (STRUT)

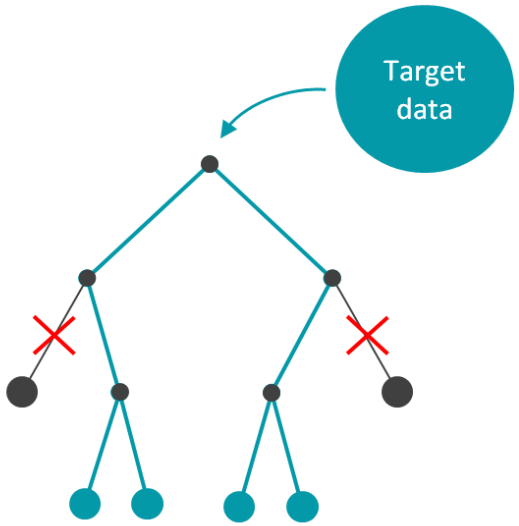


Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

Structure Expansion Reduction (SER)

Structure Transfer (STRUT)

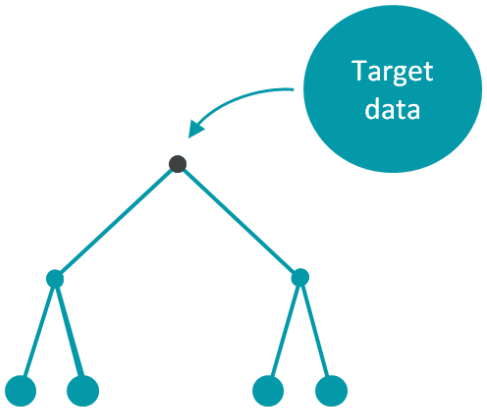


Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

Structure Expansion Reduction (SER)

Structure Transfer (STRUT)

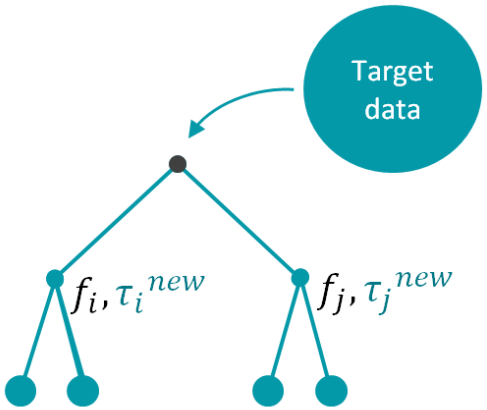


Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

Structure Expansion Reduction (SER)

Structure Transfer (STRUT)



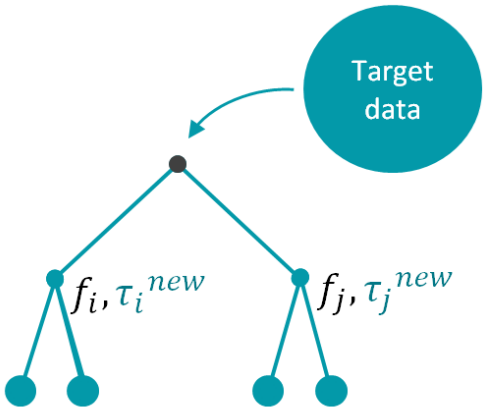
Supervised Transfer Learning on Decision Tree

A Model Transfer Learning Framework with Random Forests [Segev et al., 2017]

Structure Expansion Reduction (SER)

Partition refinement/simplification

Structure Transfer (STRUT)

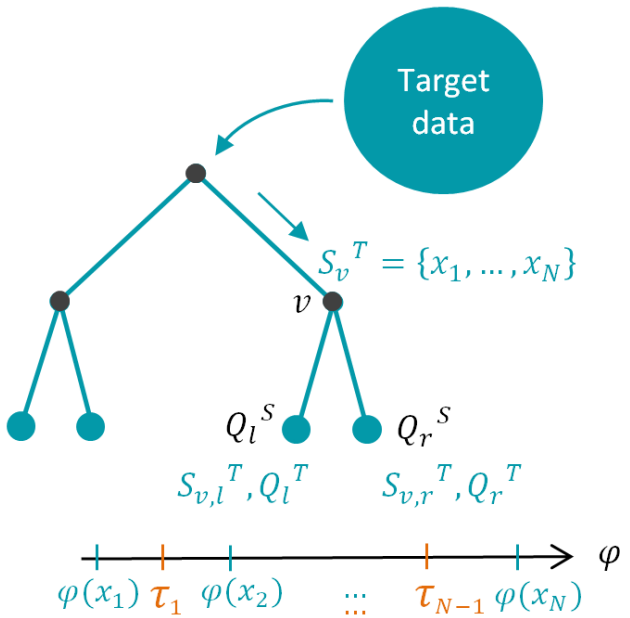


Local Drifts / Squeezes and Stretches

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 :

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

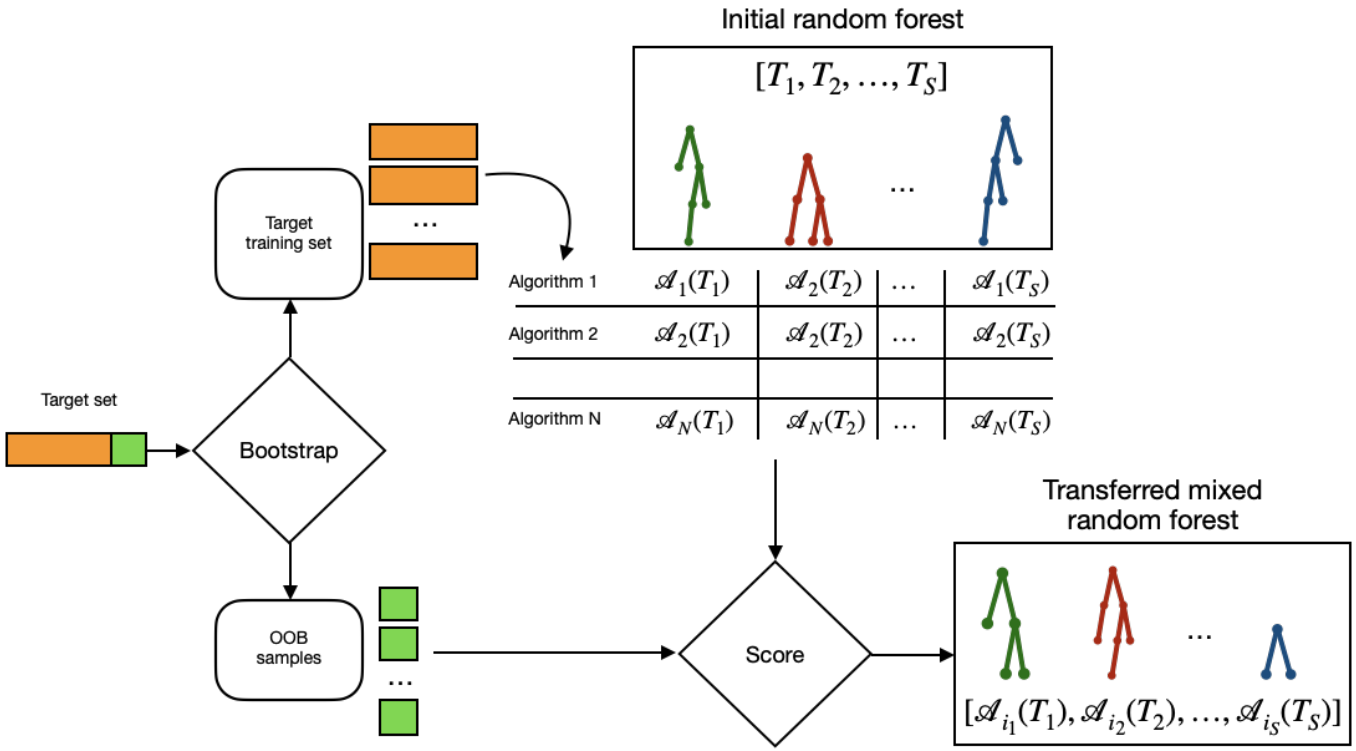
Jensen-Shannon divergence :

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

Kullback-Leibler divergence :

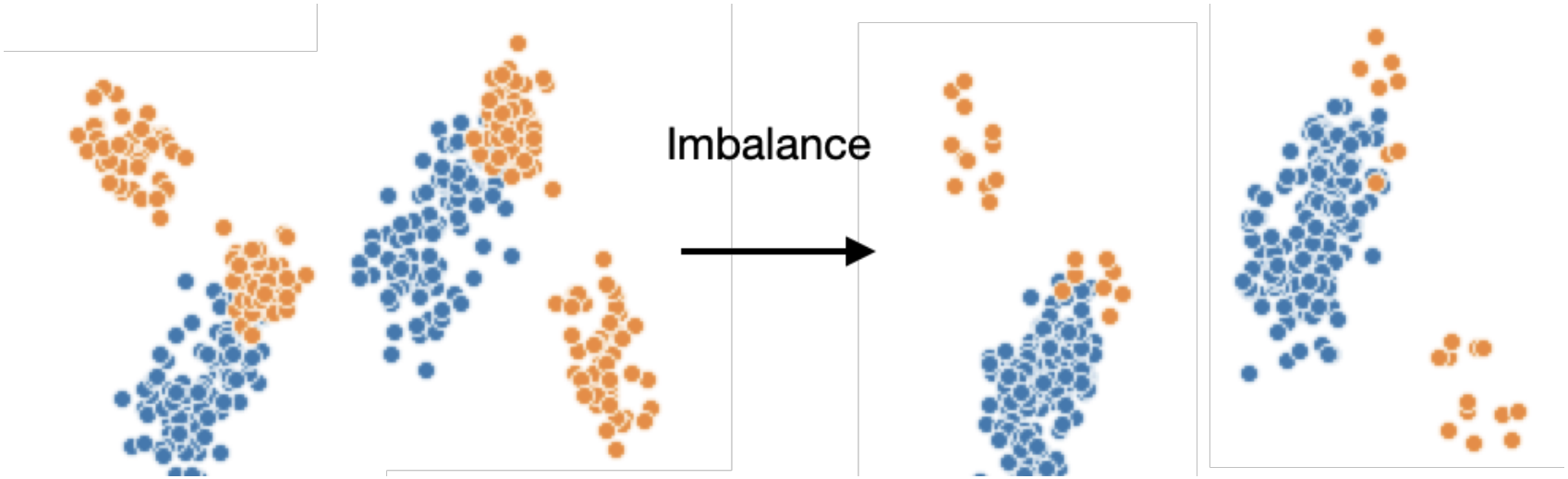
$$KL(P||Q) = \sum_{k=1}^{K} P_k \ln\left(\frac{P_k}{Q_k}\right)$$

Selective Transferred Random Forest (STRF)



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

Class Imbalance problem



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

- **Pruning** is too sensitive to data rarity
- **Divergence gain** optimization is too sensitive to 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 :

$$p^T(x|y) = p^S(x|y) \tag{1}$$

$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}, \text{with} \quad \lambda_k = \frac{P^T(y=k)}{P^S(y=k)} \tag{2}$$

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 :

$$p^T(x|y) = p^S(x|y) \tag{1}$$

$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}, \text{with} \quad \lambda_k = \frac{P^T(y=k)}{P^S(y=k)} \tag{2}$$

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 :

$$p^T(x|y) = p^S(x|y) \tag{1}$$

$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}, \text{with} \quad \lambda_k = \frac{P^T(y=k)}{P^S(y=k)} \tag{2}$$

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 :

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

$$\boxed{\overset{(2)}{\Longrightarrow} \lambda_{k_m} p^S(y=k_m|x \in l) > \lambda_k p^S(y=k|x \in l)}$$

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 :

$$p^T(x|y) = p^S(x|y) \tag{1}$$

$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}, \text{ with } \quad \lambda_k = \frac{P^T(y=k)}{P^S(y=k)} \tag{2}$$

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 :

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

$$\boxed{\overset{(2)}{\Longrightarrow} \lambda_{k_m} p^S(y=k_m|x \in l) > \lambda_k p^S(y=k|x \in l)}$$

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 :

$$p^T(x|y) = p^S(x|y) \tag{1}$$

$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}, \text{ with } \quad \lambda_k = \frac{P^T(y=k)}{P^S(y=k)} \tag{2}$$

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 :

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

$$\boxed{\overset{(2)}{\Longrightarrow} \lambda_{k_m} p^S(y=k_m|x \in l) > \lambda_k p^S(y=k|x \in l)}$$

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

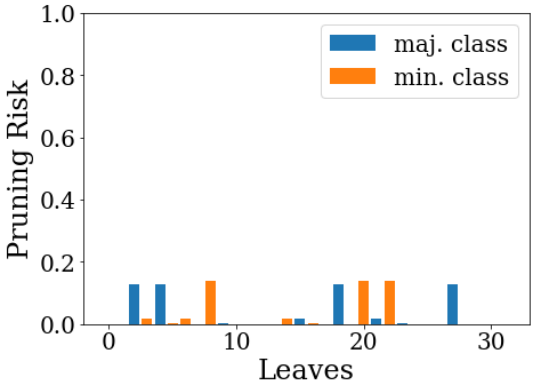
$$PRR_{n_m}(l) \overset{def}{=} p^T(\text{No data among } n_m \text{ of class } k_m \text{ class reaching } l) = p^T(x \notin l|y=k_m)^{n_m}$$

$$\boxed{\overset{(1)}{\Longrightarrow} PRR_{n_m}(l) = p^S(x \notin l|y=k_m)^{n_m}}$$

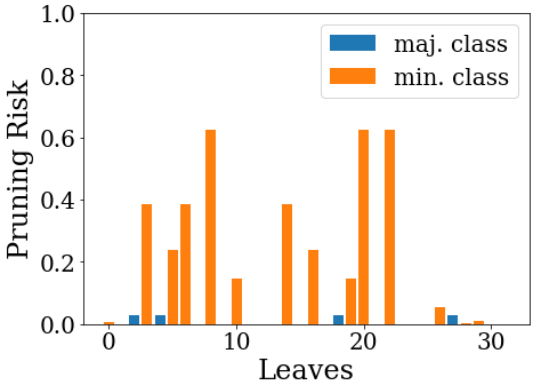
Transfer learning with Class Imbalance (Pruning risk)

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

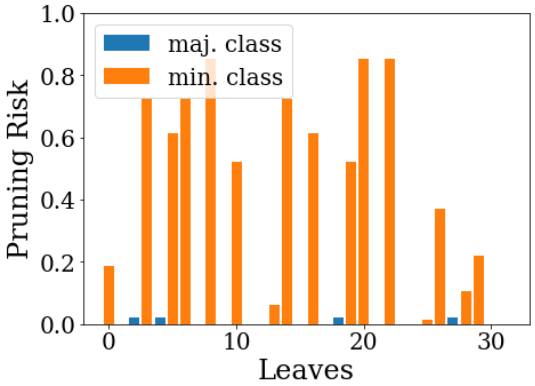
(a) Balanced / 400 instances



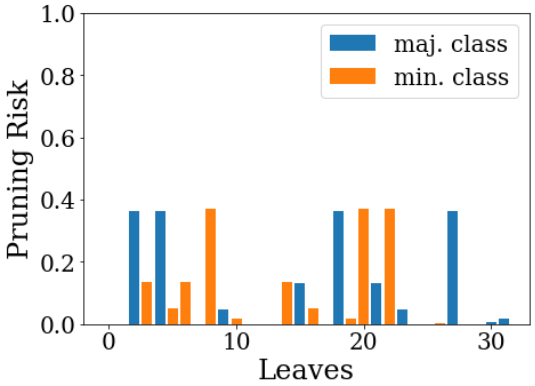
(c) Imb. (10%) / 400 instances



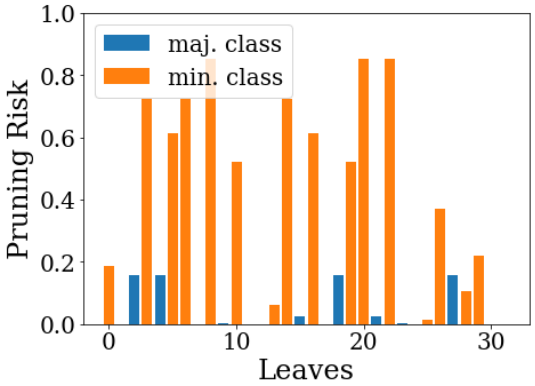
(e) Imb. (5%) / 400 instances



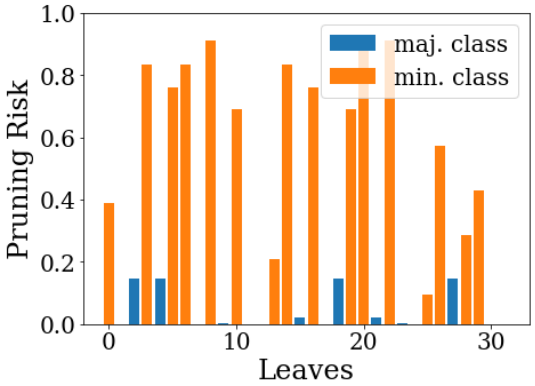
(b) Balanced / 200 instances



(d) Imb. (10%) / 200 instances

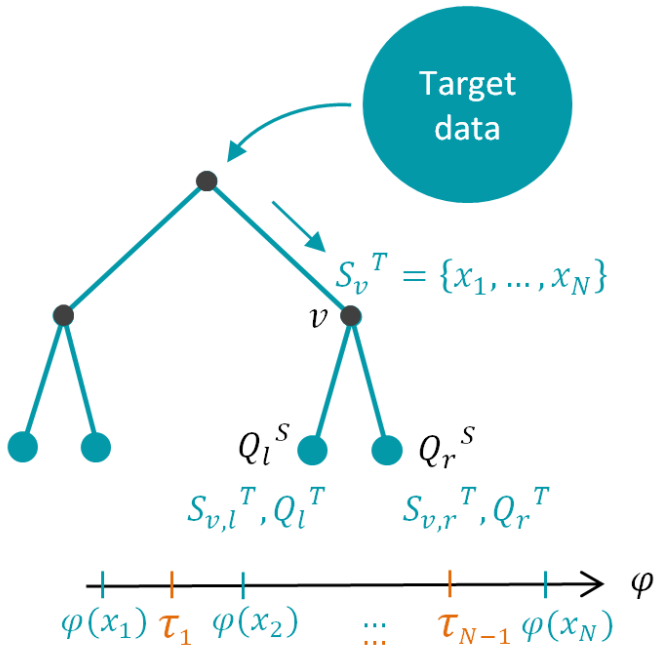


(f) Imb. (5%) / 200 instances



STRUT generalization for imbalance [Minvielle et al., 2019]

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

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

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

$STRUT(\lambda)$

Use of equation (2) :

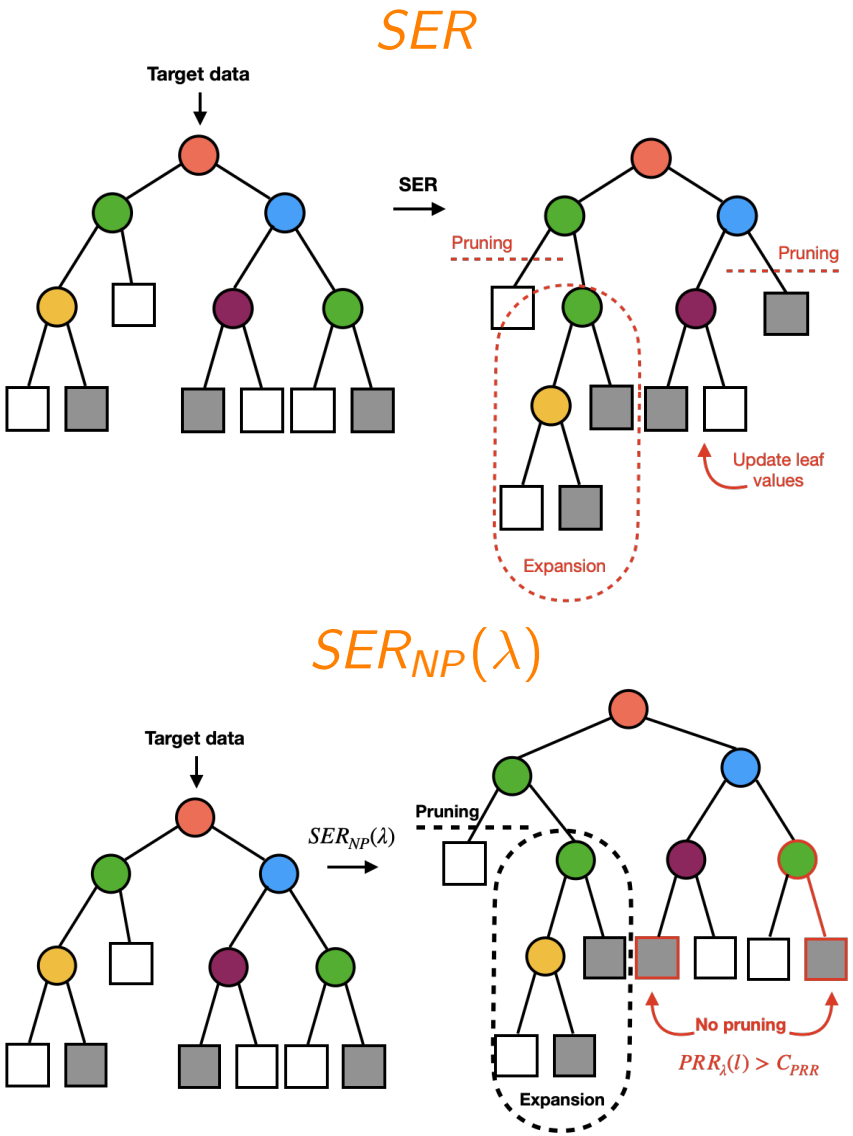
$$p^T(y=k|x) = \lambda_k \frac{p^S(y=k|x)}{\sum_{j=1}^{K} \lambda_j p^S(y=j|x)}$$

to change the source class proportions in DG :

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

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

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



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

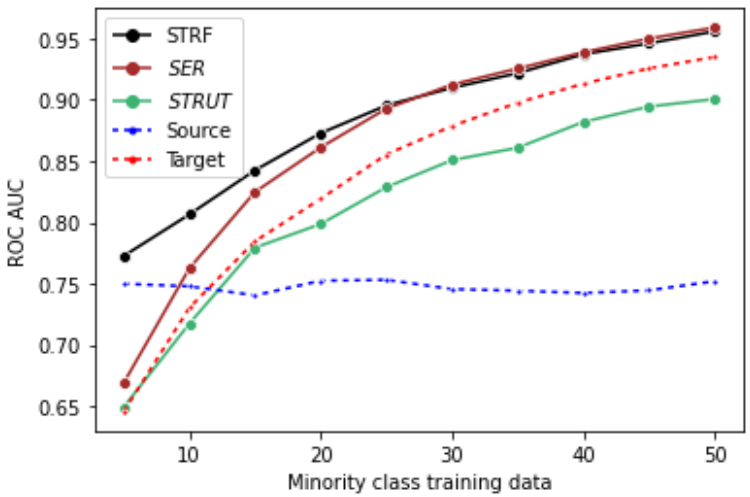
$$\lambda_{k_m} p^S(k_m | x \in l) > \lambda_k p^S(k | x \in l) : (2)$$

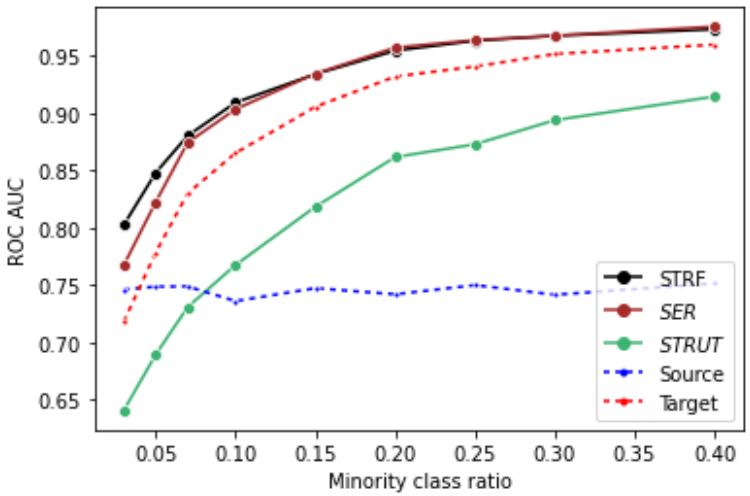
Adaptation for class imbalance :

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

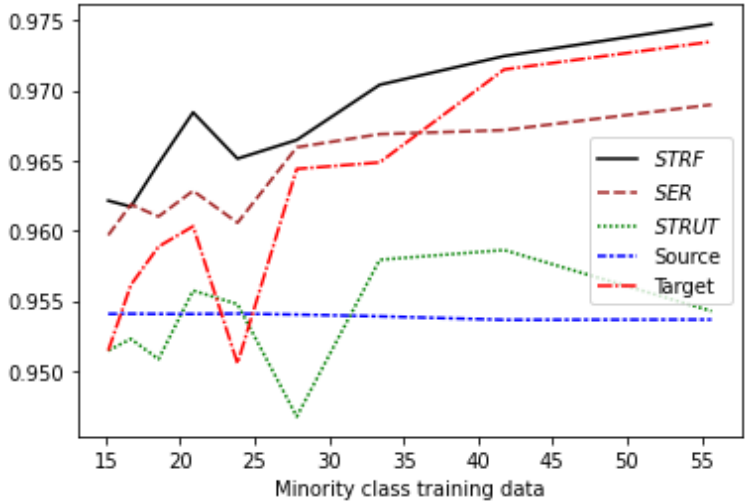
Same pruning limitation strategy on STRUT : $STRUT_{NP}(\lambda)$

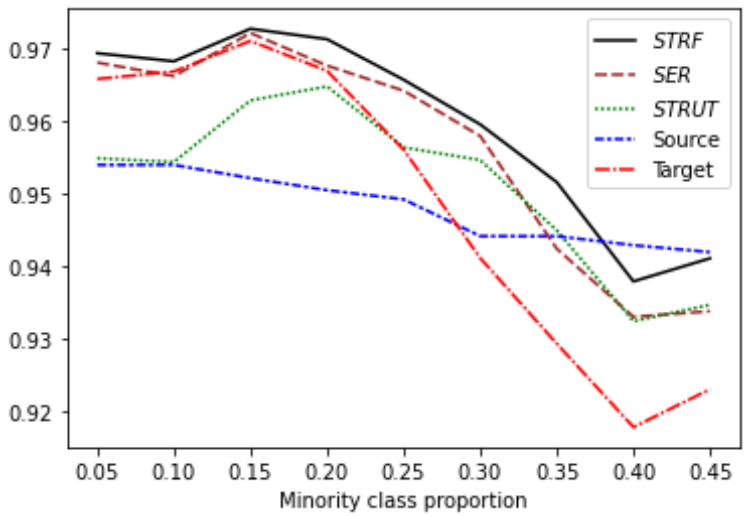
Transfer learning STRF results





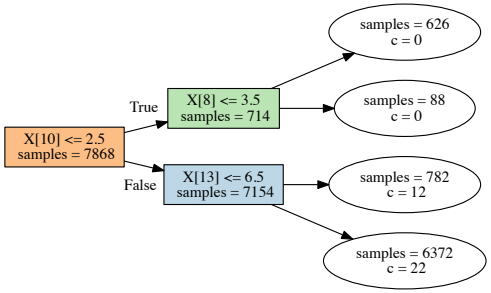
Synthetic data



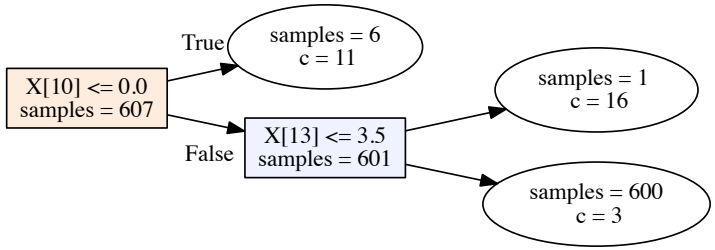


Fall detection data

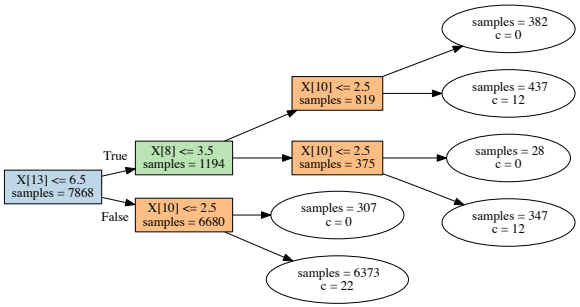
Open question



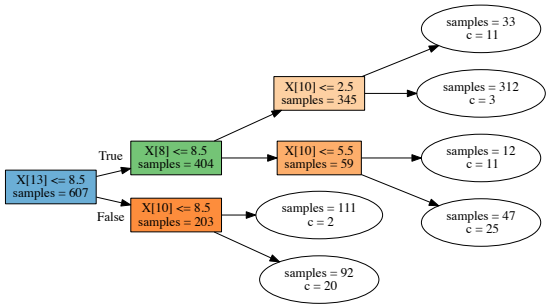
(a) T



(b) $STRUT(\mathrm{T})$



(c) T_{eq}



(d) $\mathrm{STRUT}(T_{eq})$

Figure – Output of the STRUT algorithm on two **equivalent** decision trees ($\mathcal{A}(h) \neq \mathcal{A}(h_{eq})$)

Any model-based algorithm $\mathcal{A}$ can depend on the structure of the pre-trained model !

Bibliography

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010).
A theory of learning from different domains.
Mach. Learn., 79(1-2) :151–175.

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.

Daumé III, H. (2007).
Frustratingly easy domain adaptation.
In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

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.

Truong, C., Atiq, M., Minvielle, L., Serra, R., Mougeot, M., and Vayatis, N. (2023).
A data set for fall detection with smart floor sensors.
IPOL.