




Learning Tailored Data Representations from Few Labeled Examples

Construction de Représentations de Données Adaptées dans le Cadre de peu d'Exemples Étiquetés

Léo Gautheron

Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d'Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France.

Thèse soutenue publiquement le 8 décembre 2020 devant le jury composé de :

Paulo GONÇALVES	Directeur de recherche, INRIA Rhône-Alpes	Rapporteur
Amaury HABRARD	Professeur, Université de Saint-Étienne	Co-Directeur
Emilie MORVANT	Maître de conférences, Université de Saint-Étienne	Co-Encadrante
Marc SEBBAN	Professeur, Université de Saint-Étienne	Directeur
Christine SOLNON	Professeur, INSA de Lyon	Examinatrice
Marc TOMMASI	Professeur, Université de Lille	Rapporteur

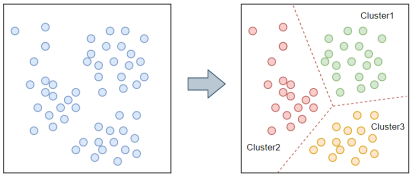
Outline

Outline

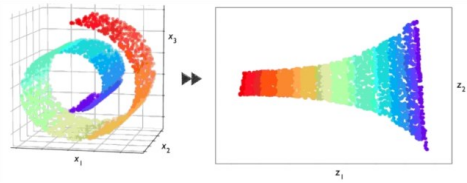
Machine learning: Learning to solve automatically tasks using data

Unsupervised Learning

Clustering

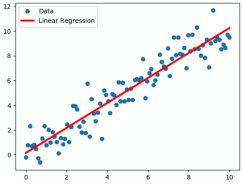


Dimensionality reduction



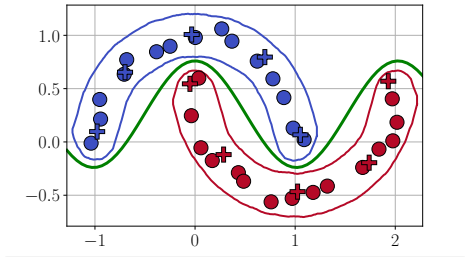
Supervised Learning

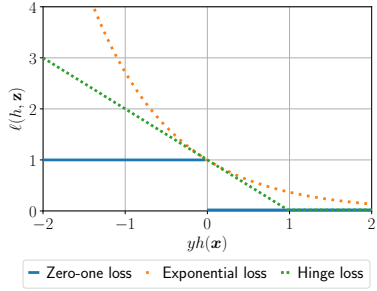
Regression



Classification

Supervised classification





Training set $S = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^m, \quad \forall i, \quad \mathbf{x}_i \in \mathbb{R}^d, \quad y_i \in \{-1, +1\}$

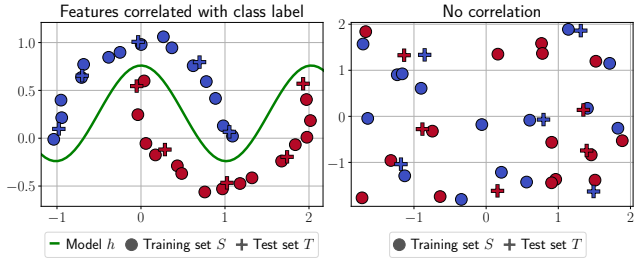
$$\underset{h}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \ell(h, \mathbf{z}_i) + \lambda Reg(h)$$

Goal: learn h consistent with S and accurate on any T drawn from $\mathcal{D}$

Features

The **quality of the features** is key in machine learning

In classification, the features need to be **correlated with the class label**



Types of features

- Observable: directly measured (*e.g.*, customer wage, patient blood pressure...)
- Handcrafted: built using expert knowledge (*e.g.*, image descriptors, feature aggregates...)
- **Latent**: automatically learned (*e.g.*, deep learning, metric learning...)

Representation learning

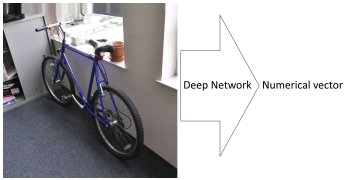
Learning latent features is advantageous

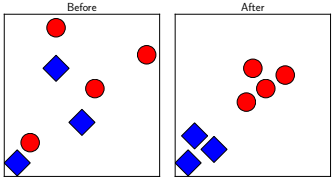
- allows to better capture correlations with labels
- replaces costly handcrafted features
- is computationally convenient to process

Deep Learning

Metric Learning

Examples:

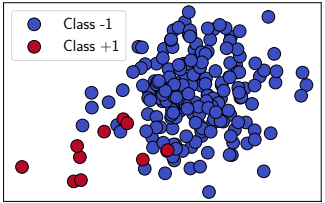




Problematic: Representation learning from few labeled examples

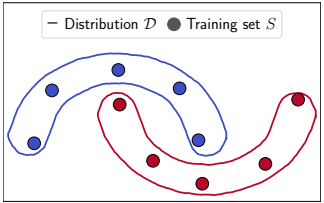
Different scenarios

Imbalanced (few positives)



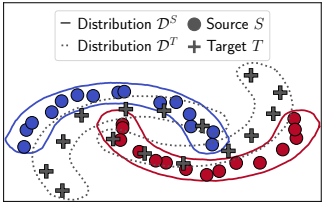
What to optimize?

Few training examples



Is it trustworthy?

Domain adaptation



How to transfer?

Drawbacks of the existing representation learning methods in these scenarios

- Favor the majority class when the data is imbalanced
- Over-fit the training data with few examples to learn from
- Build complex non interpretable latent features for domain adaptation

Contributions

Metric Learning from Imbalanced Data (CAp 2018, ICTAI 2019, PRL 2020)

- Learn a representation where the classes are treated more equally
- Derive generalization guarantees taking into account the imbalance

Ensemble Learning with RFF and Boosting (CAp 2019, CAp 2020, ECML 2020)

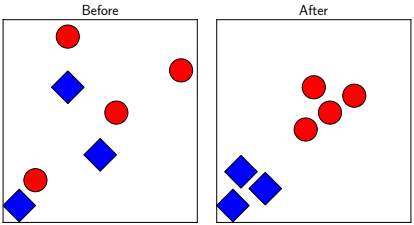
- Learn a model & a representation through boosting and kernel approximations
- Generalize well with few training examples

Representation Learning for Unsupervised Domain Adaptation (ECML 2018)

- Measure the similarity of the features between S and T with no target label
- Select the most similar features to handle domain adaptation tasks

Outline

Classical Metric Learning Approaches (LMNN, ITML, GMML, IMLS)



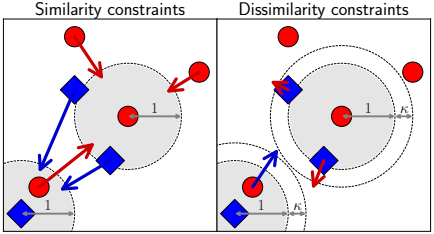
Mahalanobis distance: $d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^\top \mathbf{M} (\mathbf{x} - \mathbf{x}')}$

$$= \sqrt{(\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}')^\top (\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}')}$$

- $\mathbf{M}$ must be PSD, by Cholesky decomposition $\mathbf{M} = \mathbf{L}^\top \mathbf{L}$
- $\mathbf{L}\mathbf{x}$ is the projected vector of latent/learned features

Goal: optimize $\mathbf{M}$ under constraints so as to obtain a better k-Nearest-Neighbor classifier

Classical Metric Learning Approaches (LMNN, ITML, GMML, IMLS)



- $\mathbf{M}$ is learned so as to minimize a trade-off between two losses on dis/similar pairs

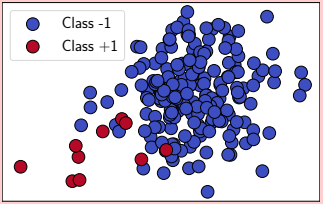
$$\min_{\mathbf{M}\succeq 0} \sum_{(\mathbf{x},\mathbf{x}')\in Sim} \ell_1(\mathbf{M},\mathbf{x},\mathbf{x}') + \sum_{(\mathbf{x},\mathbf{x}')\in Dis} \ell_2(\mathbf{M},\mathbf{x},\mathbf{x}'),$$

where $\ell_1(\mathbf{M},\mathbf{x},\mathbf{x}') = \Big[d_{\mathbf{M}}(\mathbf{x},\mathbf{x}')-1\Big]_+$ and $\ell_2(\mathbf{M},\mathbf{x},\mathbf{x}') = \Big[1+\kappa-d_{\mathbf{M}}(\mathbf{x},\mathbf{x}')\Big]_+$

- Considering all dis/similar constraints is in $O(m^2)$
- In practice for better scalability, sample randomly some constraints

Contribution

Two main drawbacks of state-of-the-art approaches with imbalanced data



1. The randomly selected pairs might not involve the minority class
2. The optimization is driven by the majority class

Contribution

New algorithmic and theoretic contributions to deal with the scarcity of positive examples

Nearest Neighbor Pair Selection

Problem: the randomly selected pairs might not involve the minority class

Aim:

- Still want to avoid to select all m^2 pairs
- But being sure to select pairs involving minority examples

Solution:

- Construct two sets of pairs Sim and Dis based on the k nearest neighbor rule
- Each example has k similar pairs in Sim and k dissimilar pairs in Dis

Reweighting the Importance of the Pairs

Problem: the optimization is driven by the majority class

Contribution: **IML** algorithm

- Split the sets of pairs Sim and Dis into four sets:
  - minority: Sim^+ and Dis^+ both with km^+ pairs
  - majority: Sim^- and Dis^- both with km^- pairs

$$\min_{\mathbf{M}\succeq 0} \sum_{(\mathbf{x},\mathbf{x}')\in Sim^+} a\ell_1(\mathbf{M},\mathbf{x},\mathbf{x}') + \sum_{(\mathbf{x},\mathbf{x}')\in Sim^-} (1-a)\ell_1(\mathbf{M},\mathbf{x},\mathbf{x}') + \sum_{(\mathbf{x},\mathbf{x}')\in Dis^+} b\ell_2(\mathbf{M},\mathbf{x},\mathbf{x}') + \sum_{(\mathbf{x},\mathbf{x}')\in Dis^-} (1-b)\ell_2(\mathbf{M},\mathbf{x},\mathbf{x}') + \lambda\|\mathbf{M}-\mathbf{I}\|_F^2$$

- Reweight the pairs by taking the imbalance into account with $a = b = \frac{m^-}{m}$

Uniform stability framework [Bousquet and Elisseeff, 2002]

Definition: Uniform stability

$$\sup_{\mathbf{z} \in S} \left| \ell(h, \mathbf{z}) - \ell(h^i, \mathbf{z}) \right| \leq \beta$$

where h is learned from S and h^i is learned after replacing the i^{th} example of S

Intuition: Learning a h^i **after a small modification** should give **almost the same model** h

Theorem

A stable algorithm in β with an upper-bounded loss by M satisfies with a probability at least $1 - \delta$ over the random draw of m examples:

$$\overbrace{R(h)}^{\text{generalization error}} \leq \overbrace{\widehat{R}(h)}^{\text{training error}} + 2\beta + \Big(4m\beta + M\Big)\sqrt{\frac{\ln(1/\delta)}{2m}}$$

Intuition: With a high probability (small δ) the **measured training error tends to the unknown generalization error** when β, M are small and m large

Generalization guarantees of the proposed **IML** algorithm

Using the uniform stability framework, we obtain the following stability constant

$$\beta = \frac{2q^2\Big(a(2\rho-1)+2(1-\rho)\Big)}{\lambda m}$$

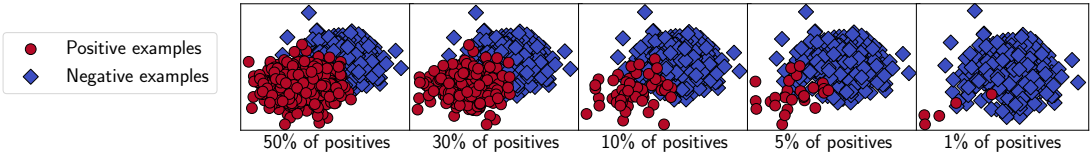
- $a \in [0,1]$ is a weight parameter with $a > 0.5$ giving the minority a higher weight
- $\rho = \frac{m^+}{m}$ is the proportion of minority examples
- q depends on an upper bound on the norm of the examples

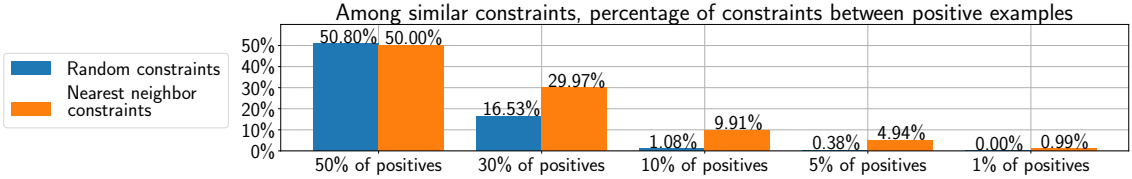
Behaviour in the presence of imbalance

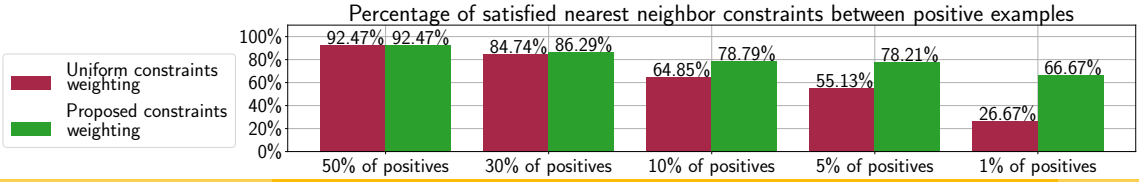
- If $a = 0.5$ and $\rho = 0.5$ (classical setting) $\rightarrow \beta = \frac{2q^2}{\lambda m}$
- If $\rho \rightarrow 0$ we have $\beta \rightarrow \frac{2q^2(-a+2)}{\lambda m}$ but $\beta \rightarrow \frac{3q^2}{\lambda m}$ without the presence of a in the bound

Conclusion: A higher a (minority class weight) reduces the negative effect of imbalance

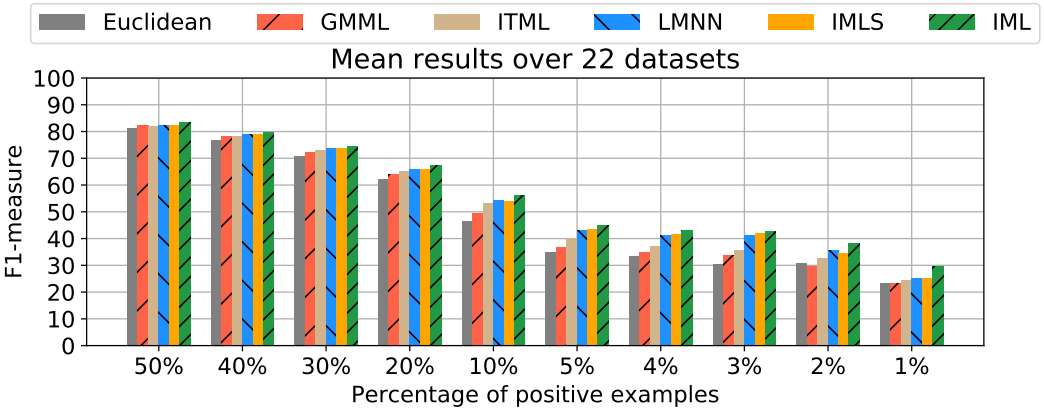
Behavior in the presence of imbalance







Comparison with state-of-the-art on 22 datasets

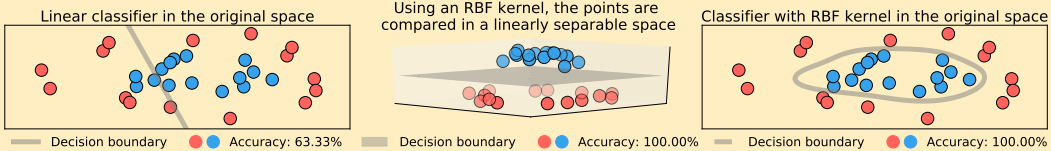


Outline

Background: Random Fourier Features [Rahimi & Recht, NeurIPS 2007]

Kernel methods

- Compare points in a higher dimensional space without explicitly projecting them



Random Fourier Features

- **Approximate** a shift-invariant kernel k in a **lower-dimensional space**
- With p the **Fourier transform** of k, and K large, we have

$$k(\mathbf{x}, \mathbf{x}') \;=\; \mathbb{E}_{\boldsymbol{\omega} \sim p} \cos\left(\boldsymbol{\omega} \cdot (\mathbf{x} - \mathbf{x}')\right) \;\simeq\; \frac{1}{K} \sum_{j=1}^{K} \cos\left(\boldsymbol{\omega}_j \cdot (\mathbf{x} - \mathbf{x}')\right)$$

- But what if k is not suited for the task at hand?

Background

PBRFF [Letarte et al. 2019]

Learn kernels based on the RFF approximation

1. Select randomly V examples called **landmarks**, learn one kernel $k_{\mathbf{x}^t}$ per landmark $\mathbf{x}^t$:

$$k_{\mathbf{x}^t}(\mathbf{x}) = \sum_{j=1}^{K} q_j^t \cos(\boldsymbol{\omega}_j^t \cdot (\mathbf{x}^t - \mathbf{x}))$$

   where q_j^t are the weights of the RFF learned using the PAC-Bayesian theory
2. Train a linear model in the following mapping:

$$\psi(\mathbf{x}) = (k_{\mathbf{x}^1}(\mathbf{x}), \ldots, k_{\mathbf{x}^V}(\mathbf{x}))$$

Drawbacks of the approach

- How to select good landmarks?
- We can only train the model after learning the representation

Contribution

We learn at the same time the model and the landmarks through gradient boosting

Two algorithms are proposed for this purpose:

Algorithm **GBRFF1**:

- the base learner is the kernel proposed by Letarte et al
- the landmarks and kernels are learned through the gradient boosting exponential loss

Algorithm **GBRFF2**:

- a speedup using a single RFF instead of many without degrading the performances
- improved performances by learning the random part of the RFF

Both of them are based on a **gradient boosting** algorithm

Gradient Boosting [Friedman 2001]

Build classifiers making predictions based on a weighted sum of predictors:

$$\text{sign}\left(H^0(\mathbf{x}) + \sum_{t=1}^{V} \alpha^t h^t(\mathbf{x})\right)$$

where

- H^0 an initial predictor minimizing the exponential loss noted ℓ
- h^t a regression model trained to fit the residuals $\tilde{y}_i$

$$\tilde{y}_i = -\frac{\partial \ell(y_i, H^t(\mathbf{x}_i))}{\partial H^t(\mathbf{x}_i)}$$

- α^t optimal weight of h^t minimizing ℓ
- V number of iterations

Contribution: GBRFF1

A first version close to PBRFF: learn a model and the landmarks at the same time

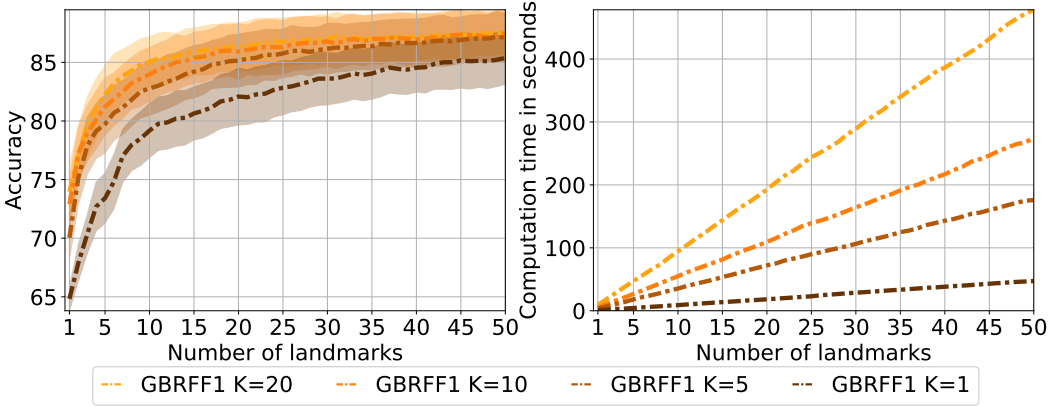
The regression model h^t is the kernel $k_{\mathbf{x}^t}$ of Letarte et al

$$h^t(\mathbf{x}) = \sum_{j=1}^{K} q_j^t \cos(\boldsymbol{\omega}_j^t \cdot (\mathbf{x}^t - \mathbf{x}))$$

At each iteration

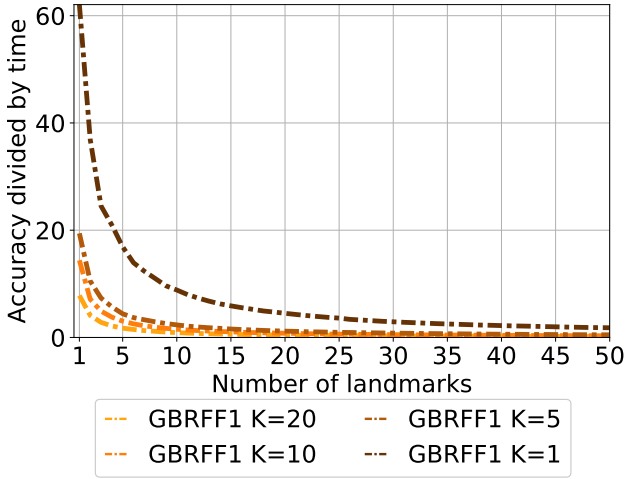
- draw $\boldsymbol{\omega}_j^t \sim \mathcal{N}(0, 2\gamma)^d$ which is the Fourier transform of the RBF kernel defined as $k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$
- the landmark $\mathbf{x}^t$ is learned to minimize the exponential loss at iteration t
- the RFF weights q_j^t are learned using **PBRFF**

Experiment: Analyzing GBRFF1



As expected, higher values of K give better performances

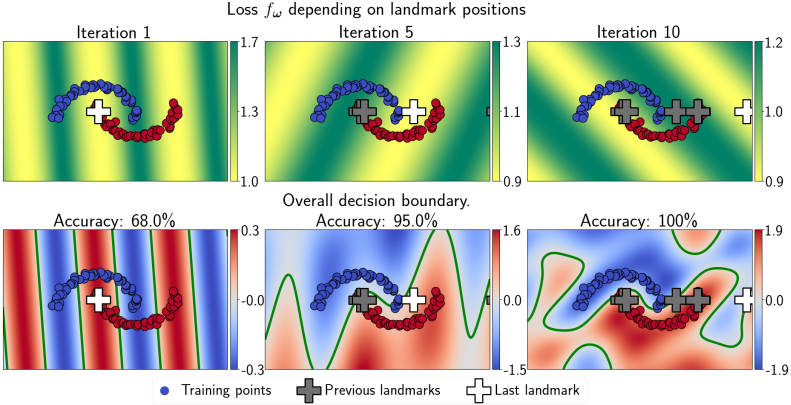
Experiment: Improving GBRFF1 using $K = 1$



Higher K gives better performances, but $K=1$ gives the best accuracy/time trade-off

Contribution: GBRFF2

The regression model becomes $h^t(\mathbf{x}) = \cos(\boldsymbol{\omega}^t \cdot \mathbf{x} - b^t)$. Using the cosine periodicity with $K{=}1 \rightarrow$ cheaper landmark learning: learn a scalar b^t instead of a vector $\mathbf{x}^t$



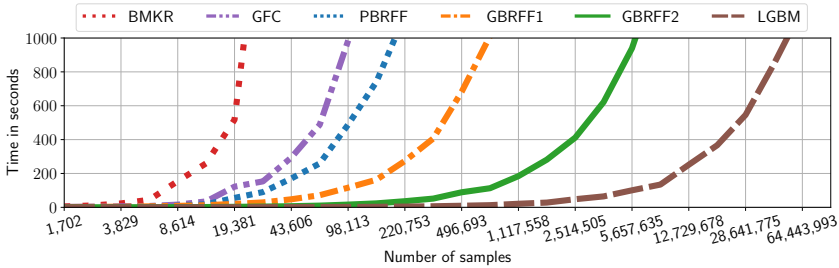
After (1) drawing $\boldsymbol{\omega}^t$ and (2) learning b^t, then (3) fine-tune $\boldsymbol{\omega}^t$ to improve the performances

Comparison to the state-of-the-art

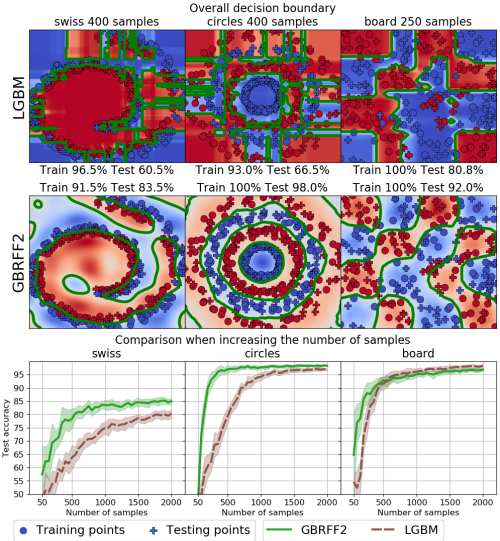
- LGBM is a state-of-the-art gradient boosting method using trees as base predictors
- BMKR is a Multiple Kernel Learning method using SVR inside gradient boosting
- GFC is a greedy feature construction method based on functional gradient descent

Mean results over 16 datasets

Dataset	BMKR	GFC	PBRFF	GBRFF1	LGBM	GBRFF2
Mean	88.4 ± 2.1	85.7 ± 2.0	87.9 ± 2.0	87.9 ± 2.0	89.0 ± 2.1	$\mathbf{89.1} \pm 2.0$
Average Rank	2.88	4.94	3.75	3.81	3.44	**2.19**

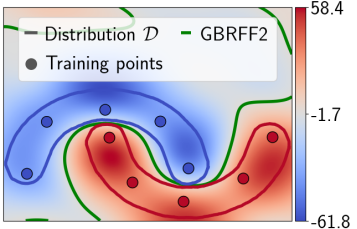


Toy illustrations



Comparison of LGBM and GBRFF2 on 3 toy datasets

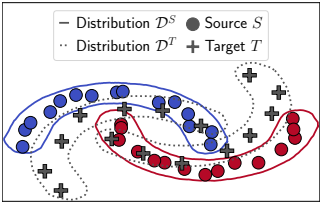
GBRFF2 generalizes well with few training examples



Outline

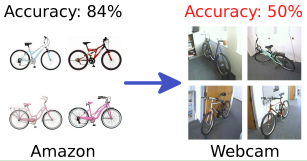
Background: Domain Adaptation (DA)

Learning from S coming from a distribution $\mathcal{D}^S$, deploy on T with a different distribution $\mathcal{D}^T$



Issue: Few, or even no label available in T

A model learned on S may perform poorly on T due to the distribution shift



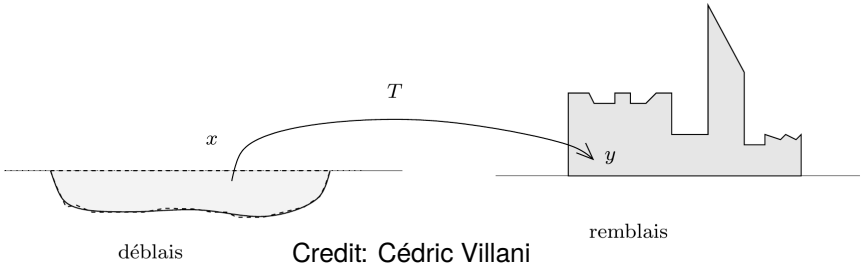
Domain adaptation seeks to deal with the shift and obtain a good model on T using S

Background: discrete optimal transport [Monge 1781, Kantorovich 1942]

Find the optimal coupling $\boldsymbol{\gamma}^*$ that aligns $\boldsymbol{S} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{T} \in \mathbb{R}^{n \times d}$ as:

$$\boldsymbol{\gamma}^* = \underset{\boldsymbol{\gamma} \in \Pi(\widehat{\boldsymbol{\mathcal{D}}}^S, \widehat{\boldsymbol{\mathcal{D}}}^T)}{\operatorname{argmin}} \langle \boldsymbol{\gamma}, \boldsymbol{C} \rangle_F$$

- with $\widehat{\boldsymbol{\mathcal{D}}}^S$, $\widehat{\boldsymbol{\mathcal{D}}}^T$ empirical distributions on $\boldsymbol{S}$, $\boldsymbol{T}$ (default: uniform distributions)
- $\boldsymbol{C} \in \mathbb{R}^{m \times n}$ transport cost from $\boldsymbol{S}$ to $\boldsymbol{T}$ (default: Euclidean distance)
- $\Pi(\widehat{\boldsymbol{\mathcal{D}}}^S, \widehat{\boldsymbol{\mathcal{D}}}^T) = \{\boldsymbol{\gamma} \in \mathbb{R}_+^{m \times n} | \boldsymbol{\gamma}\mathbf{1} = \widehat{\boldsymbol{\mathcal{D}}}^S, \boldsymbol{\gamma}^\top \mathbf{1} = \widehat{\boldsymbol{\mathcal{D}}}^T\}$

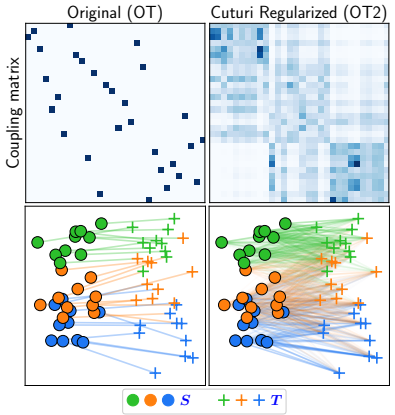


Credit: Cédric Villani

Regularized optimal transport [Cuturi 2013]

Fast computation (Sinkhorn-Knopp algorithm) & γ smooth with an entropy regularization

$$\gamma^* = \underset{\gamma \in \Pi(\widehat{\mathcal{D}}^S, \widehat{\mathcal{D}}^T)}{\operatorname{argmin}} \langle \gamma, \boldsymbol{C} \rangle_F - \frac{1}{\lambda} E(\gamma)$$



Our contribution: main idea

Motivation

- Datasets with **many features** with some of them **too domain specific** (bag-of-words, image descriptors, word embeddings...)
- If $\boldsymbol{S}$ and $\boldsymbol{T}$ are described by the same features, reduce the discrepancy by **discarding the most shifted/specific features**

Proposed idea

1. Find a **coupling** $\boldsymbol{\gamma}^* \in \mathbb{R}_+^{d \times d}$ **between the features** of $\boldsymbol{S}$ and $\boldsymbol{T}$
   The larger γ_{ii}^*, the most similar the feature number i between $\boldsymbol{S}$ and $\boldsymbol{T}$
2. Keep the most similar features sorted by their similarity given in the diagonal of $\boldsymbol{\gamma}^*$

Step 1: Sample selection

Problem: Computing $\boldsymbol{C} \in \mathbb{R}^{d \times d}$ requires the same number of examples in $\boldsymbol{S}$ and $\boldsymbol{T}$

Solution: Find a matching between $\boldsymbol{S}$ and $\boldsymbol{T}$, keep only the most correlated instances

Algorithm 1: Sample selection in target domain

Input : $\boldsymbol{S} \in \mathbb{R}^{m \times d}$, $\boldsymbol{T} \in \mathbb{R}^{n \times d}$

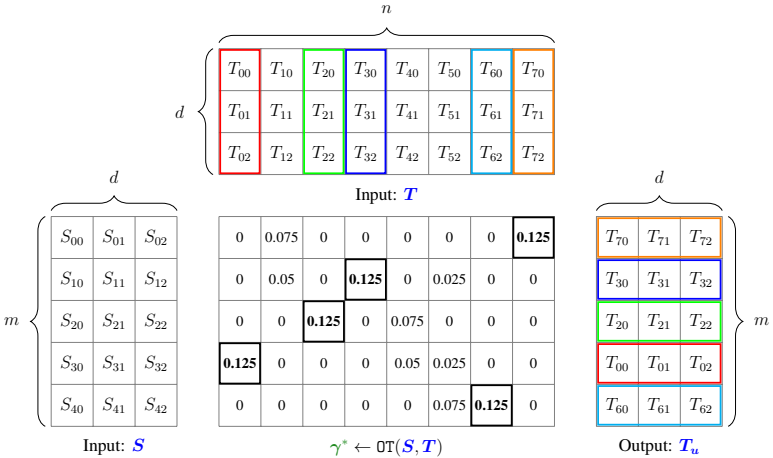
Output : $\boldsymbol{T_u} \in \mathbb{R}^{m \times d}$

$\mathtt{S} = \mathtt{zscore}(\mathtt{S})$

$\mathtt{T} = \mathtt{zscore}(\mathtt{T})$

$\boldsymbol{\gamma}^* \leftarrow \mathtt{OT}(\mathtt{S}, \mathtt{T})$

$\mathtt{T_u} \leftarrow \{\mathtt{x_j} \in \mathtt{T} | \mathtt{j} = \underset{\mathtt{i}=1,\ldots,\mathtt{m}}{\mathrm{argmax}} \gamma^*_{\mathtt{ij}}\}$



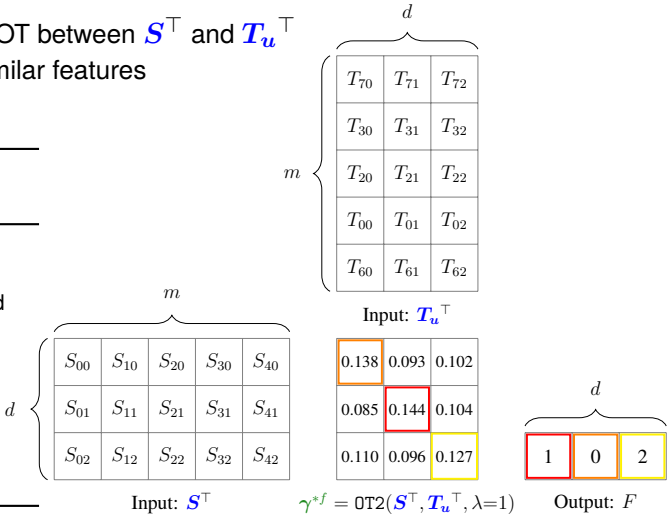
Step 2: Feature ranking

Compute γ^{*f} as entropy-regularized OT between $\boldsymbol{S}^\top$ and $\boldsymbol{T_u}^\top$
Sort diag(γ^{*f}) to identify the most similar features

Algorithm 2: Feature ranking for domain adaptation

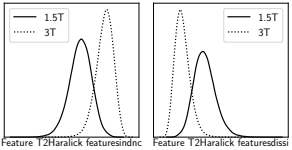
Input : $\boldsymbol{S} \in \mathbb{R}^{m \times d}, \boldsymbol{T} \in \mathbb{R}^{n \times d}$
Output : List of features F ordered by decreasing similarity between $\boldsymbol{S}$ and $\boldsymbol{T}$

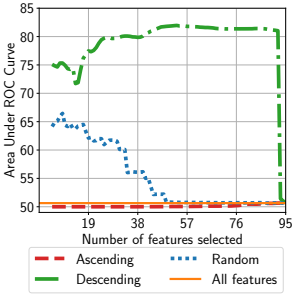
$\mathtt{T_u} \leftarrow \mathtt{Algorithm1}(\boldsymbol{S}, \boldsymbol{T})$
$\mathtt{S}^\top = \mathtt{zscore}(\mathtt{S}^\top)$
$\mathtt{T_u}^\top = \mathtt{zscore}(\mathtt{T_u}^\top)$
$\gamma^{*\mathtt{f}} = \mathtt{OT2}(\mathtt{S}^\top, \mathtt{T_u}^\top, \lambda{=}1)$
$\mathtt{F}{=}\ \mathtt{argSortDesc}(\{\gamma^{*\mathtt{f}}_{\mathtt{ii}} | \mathtt{i} \in [0, \mathtt{d}[\})$

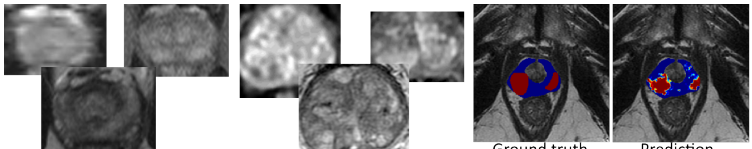


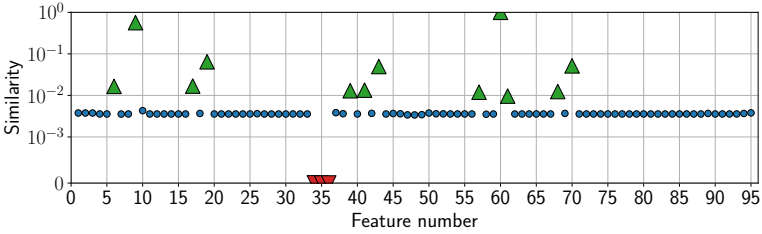
Real-world medical application: Prostate cancer mapping

Data: Medical images from two scanners 1.5T $\boldsymbol{S}$ and 3T $\boldsymbol{T}$; **95** handcrafted descriptors per voxel (3D pixels).
Each voxel is **Cancer** or **Non cancer**. **Problem**: Some features are shifted between $\boldsymbol{S}$ and $\boldsymbol{T}$







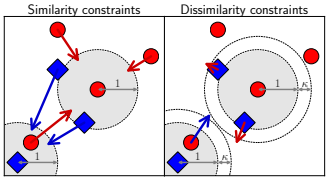


Removing the 3 most shifted features gives an AUC of 81% instead of 50% with all the features

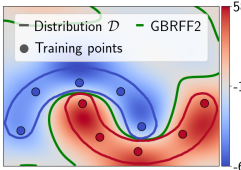
Outline

Conclusion

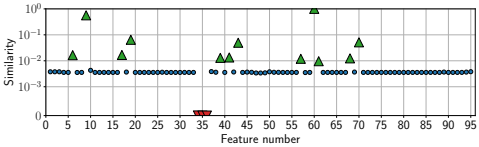
Metric Learning from Imbalanced Data
(CAp 2018, ICTAI 2019, PRL 2020)
Learn a representation in the presence of few positive examples with generalization guarantees



Ensemble Learning with RFF and Boosting
(CAp 2019, CAp 2020, ECML 2020)
Learn the model & the representation together, generalizes well with few examples



Representation Learning for Unsupervised DA
(ECML 2018)
Select similar features for domain adaptation with no target labeled example



Source code: `https://leogautheron.github.io`

Perspectives: scalability

- The ML and DA contributions make use of **costly pairwise distance matrices**
- For the ML contribution, a **batch optimization** might solve the issue. But enforcing the PSD constraint in such a context might not be trivial
- The RFF contribution is rather fast, but **gradient boosting tricks** might speed-up further the computations (careful **selection of the residuals fitted at each iteration**)
- For the DA contribution, the resulting **coupling is itself costly to store**. A possibility to reduce this storage issue could be to learn the parameters of a **continuous function that approximates the coupling values**

Perspectives: Capture non linearity

- The ML contribution **learns new features as a linear combination of the existing ones**
- The use of RFF as a pre-processing could easily induce a **kernelized metric learning method**, but another strategy by **learning at the same time the RFF and the linear combinations** might induce even stronger representations
- For the DA contribution, the optimal transport cost was fixed to the Euclidean distance. Learning a transport cost through RFF combinations might allow to capture non-linearity between features

Perspectives: Generalization guarantees

- There are **very few** metric learning **theoretical results in the presence of imbalance**. One might explore other frameworks in this **non iid setting** such as the algorithmic robustness or the U-Statistics
- The proposed Gradient Boosting method does not come with any theory, and **giving guarantees** in this context might **show its generalization capability**

Thank you for your attention

Publication in Journal

Léo Gautheron, Emilie Morvant, Amaury Habrard and Marc Sebban. Metric Learning from Imbalanced Data with Generalization Guarantees. In *Pattern Recognition Letters*, volume 133, pages 298-304. 2020

Publications in International Conferences

Léo Gautheron, Pascal Germain, Amaury Habrard, Guillaume Metzler, Emilie Morvant, Marc Sebban and Valentina Zantedeschi Landmark-based Ensemble Learning with Random Fourier Features and Gradient Boosting. In *European Conference on Machine Learning & Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, 2020

Léo Gautheron, Amaury Habrard, Emilie Morvant and Marc Sebban. Metric Learning from Imbalanced Data. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, Portland, United States

Léo Gautheron, Ievgen Redko and Carole Lartizien. Feature Selection for Unsupervised Domain Adaptation using Optimal Transport. In *European Conference on Machine Learning & Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, 2018, Dublin, Ireland

Communications in National Conferences

Léo Gautheron, Pascal Germain, Amaury Habrard, Guillaume Metzler, Emilie Morvant, Marc Sebban and Valentina Zantedeschi. Apprentissage d'ensemble basé sur des points de repère avec des caractéristiques de Fourier aléatoires et un renforcement du gradient. In *Conférence sur l'Apprentissage automatique (CAp)*, 2020

Léo Gautheron, Pascal Germain, Amaury Habrard, Gaël Letarte, Emilie Morvant, Marc Sebban and Valentina Zantedeschi. Revisite des "random Fourier features" basée sur l'apprentissage PAC-Bayésien via des points d'intérêts. In *Conférence sur l'Apprentissage automatique (CAp)*, 2019, Toulouse, France

Léo Gautheron, Amaury Habrard, Emilie Morvant and Marc Sebban. Apprentissage de métrique pour la classification supervisée de données déséquilibrées. In *Conférence sur l'Apprentissage automatique (CAp)*, 2018, Rouen, France

Experimental setup

Data

- 22 binarized datasets from UCI repository
- Artificially increase/decrease class imbalance in each datasets

Learning

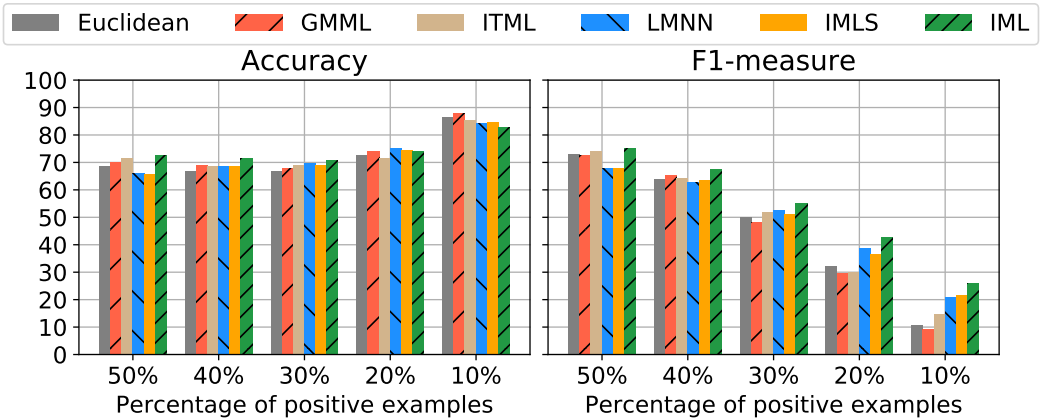
- 20 random stratified train/test splits
- Learn metric using train, project train and test in new space
- Use 3 nearest neighbor classifier

Evaluation

- Mean test result over 20 splits
- F1-measure suited for imbalance data

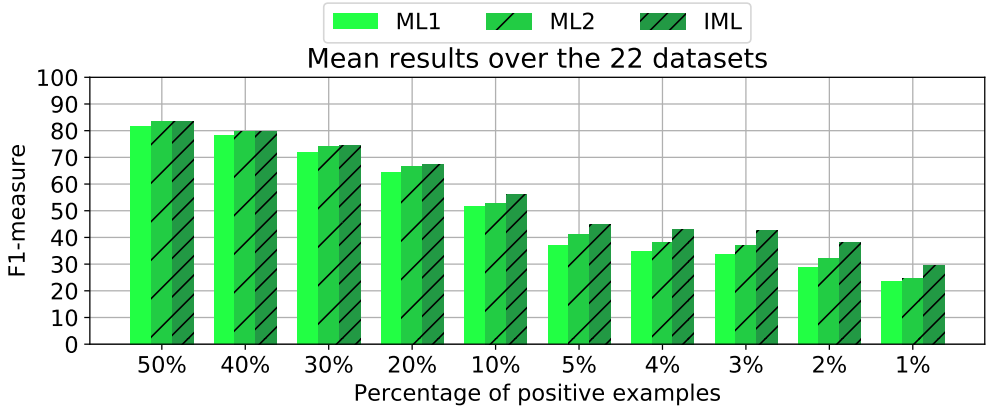
Comparison with state-of-the-art on the SPECTFHEART dataset

Name	m	d	c	$Label$	m^+	%
spectfheart	267	44	2	0	55	20.60%

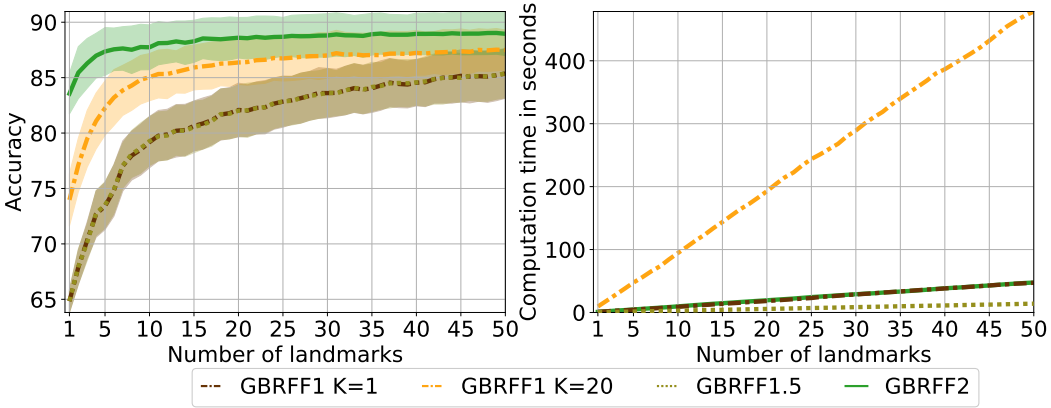


Analyzing why our method is better on imbalanced data

ML1	classical loss	random set of pairs
ML2	classical loss	k similar & k dissimilar pairs per example
IML	proposed loss reweighting	k similar & k dissimilar pairs per example



Experiment: GBRFF1 vs GBRFF2



- GBRFF1.5 identical to GBRFF1 $K = 1$ except uses cheaper landmark learning
- Learning ω in GBRFF2 improves performances of GBRFF1.5.
  Better performances and faster than GBRFF1 with $K = 20$ RFF

Office-Caltech benchmark data set

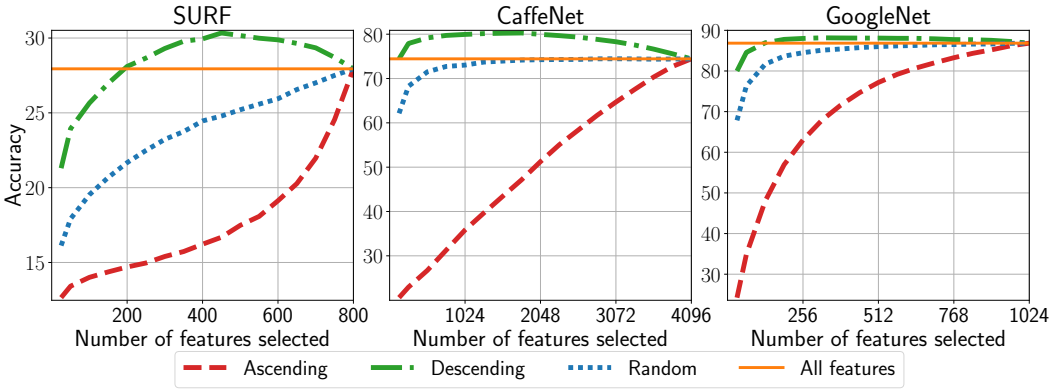
4 datasets of images sharing the same 10 classes but having their specificities.

Goal: learn a classifier on a domain to classify images of another domain

Results: report mean over 12 domain adaptation pairs

Three types of features: SURF image descriptors (d=800), and latent features from last hidden layer of two Deep Neural networks: CaffeNet (d=4096) and GoogleNet (d=1024).

Office-Caltech: accuracy results



Office-Caltech: running time speed-up

Use our method as pre-processing step of state-of-the-art domain adaptation algorithms:

- **CORAL** [Sun et al., 2016]
- **TCA** [Pan et al., 2011]
- **SA** [Fernando et al., 2013]
- **OT3** [Courty et al., 2014]

Method	↘512		↘1024		↘2048		4096	
No adapt.	79.2±2.2	0.00s	79.9±2.3	0.00s	80.0±2.2	0.00s	74.4±3.0	0.00s
CORAL	80.5±1.8	110.43s	80.8±1.9	587.69s	80.4±1.7	3996.20s	80.1±1.7	29930.39s
SA	81.8±2.0	13.25s	82.5±1.8	32.09s	82.9±1.7	66.71s	83.0±1.7	169.71s
TCA	83.5±2.2	221.08s	85.0±1.9	223.62s	85.8±1.8	229.48s	85.9±1.7	242.71s
OT3	84.2±2.4	19.50s	86.7±1.9	31.76s	88.8±1.5	54.07s	88.8±1.4	97.47s

+ Important speed-up with almost the same classification results!

Experiments on digit recognition and textual product reviews

Digits: USPS/MNIST images, 10 classes of digits, use raw pixel values as features, mean results over 2 adaptation pairs

Amazon review: Textual reviews of four product types, each review classified as positive or negative, word embedding with size 5000, mean results over 12 adaptation pairs