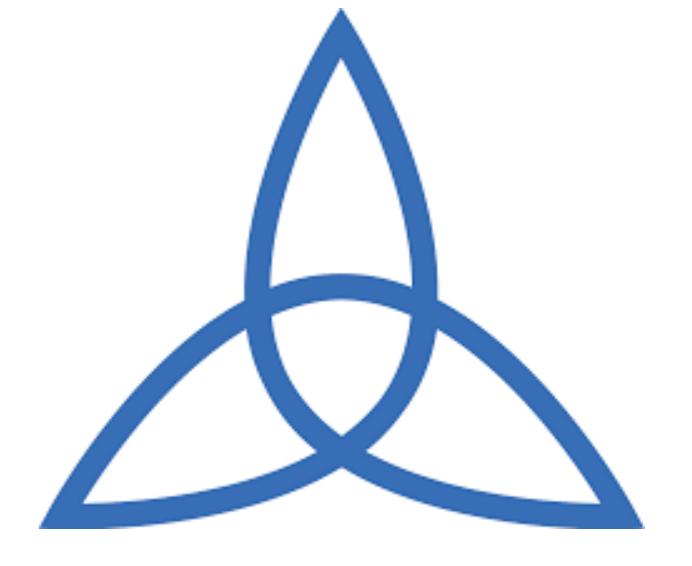


Easily Interpretable, Non-parametric Sample Transformation for Classification



Cédric Dubois¹, Jean-Olivier Irisson², Éric Debrouve¹

¹ Université Côte d'Azur, CNRS, Inria, Equipe Morpheme, Laboratoire I3S, Sophia Antipolis, France.

² Sorbonne Université, Equipe COMPLEX, Laboratoire LOV, Institut IMEV, Villefranche-sur-mer, France.

Contact: cedric.dubois@univ-cotedazur.fr

This work has been supported by the project WWW-PIC, funded by the Belmont Forum, managed by the ANR.

Nancy
06-09/09/2022

1 INTRODUCTION

Goal

- Multi-class, non-linear classification
- ANNs inspiration
- Low energy consumption
- Easy to tune
- Fast

Neural Networks are regressors

Artificial Neural Networks (ANNs) are not classifiers, but **regressors** from \mathbb{R}^d to \mathbb{R}^p . The actual decision rule is taken by the nearest-target classifier.

2 APPROACH

ANNs

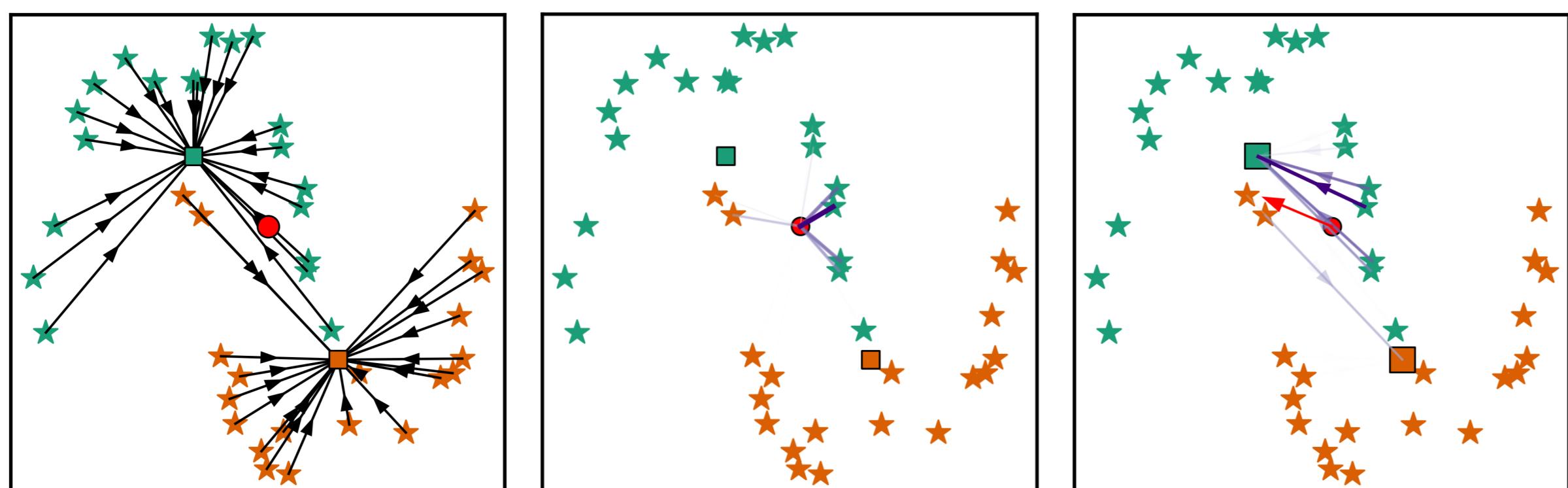
- 1 Transform samples (**regression**) :
 - From input space \mathbb{R}^d to target space \mathbb{R}^p
 - Targets defined as the canonical basis
 - Parametric, with optimization
 - 2 Take arg max (**classification**) :
 - Nearest-target classifier
- $$\hat{y} = \arg \min_y \|f_\theta(\mathbf{x}) - \mathbf{t}_y\|^2$$

Proposed Method

- 1 Transform samples (**regression**) :
 - Input and target space in \mathbb{R}^d
 - Targets defined as the class-barycenter
 - Explicit form, no optimization needed
- 2 Use simple classifier (**classification**) :
 - Nearest-target classifier
 - Any classifier

3 METHOD

Sample Transformation



Linear combination of known translations

$$\varphi_\gamma(\mathbf{u}) = \mathbf{u} + \phi_\gamma(\mathbf{u}) = \mathbf{u} + \sum_{i=1}^n w_\gamma(\mathbf{t}_{y_i} - \mathbf{x}_i)$$

where the weight function w_γ is defined such that

$$w_\gamma = \frac{K_\gamma(\mathbf{u} - \mathbf{x}_i)}{\sum_{j=1}^n K_\gamma(\mathbf{u} - \mathbf{x}_j)}$$

where K_γ is a symmetric kernel monotonically decreasing with the L2 norm of its argument and fulfilling the usual properties (real-valued, non-negative, and integrable with a unit integral).

Kernel definition

we can define a kernel using the transformation

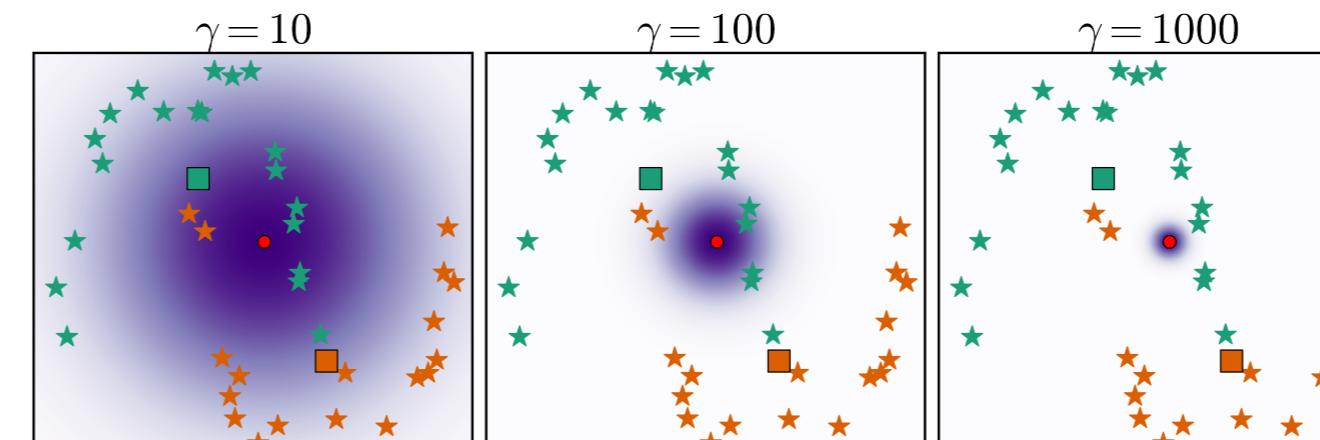
$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi_\gamma(\mathbf{x}_i), \varphi_\gamma(\mathbf{x}_j) \rangle.$$

Simple Classification

Classification after transformation:

- Nearest-Target classifier
 - Any classifier
- $$\hat{y} = \arg \min_y \|\varphi_\gamma(\mathbf{u}) - \mathbf{t}_y\|^2$$

Parameter γ

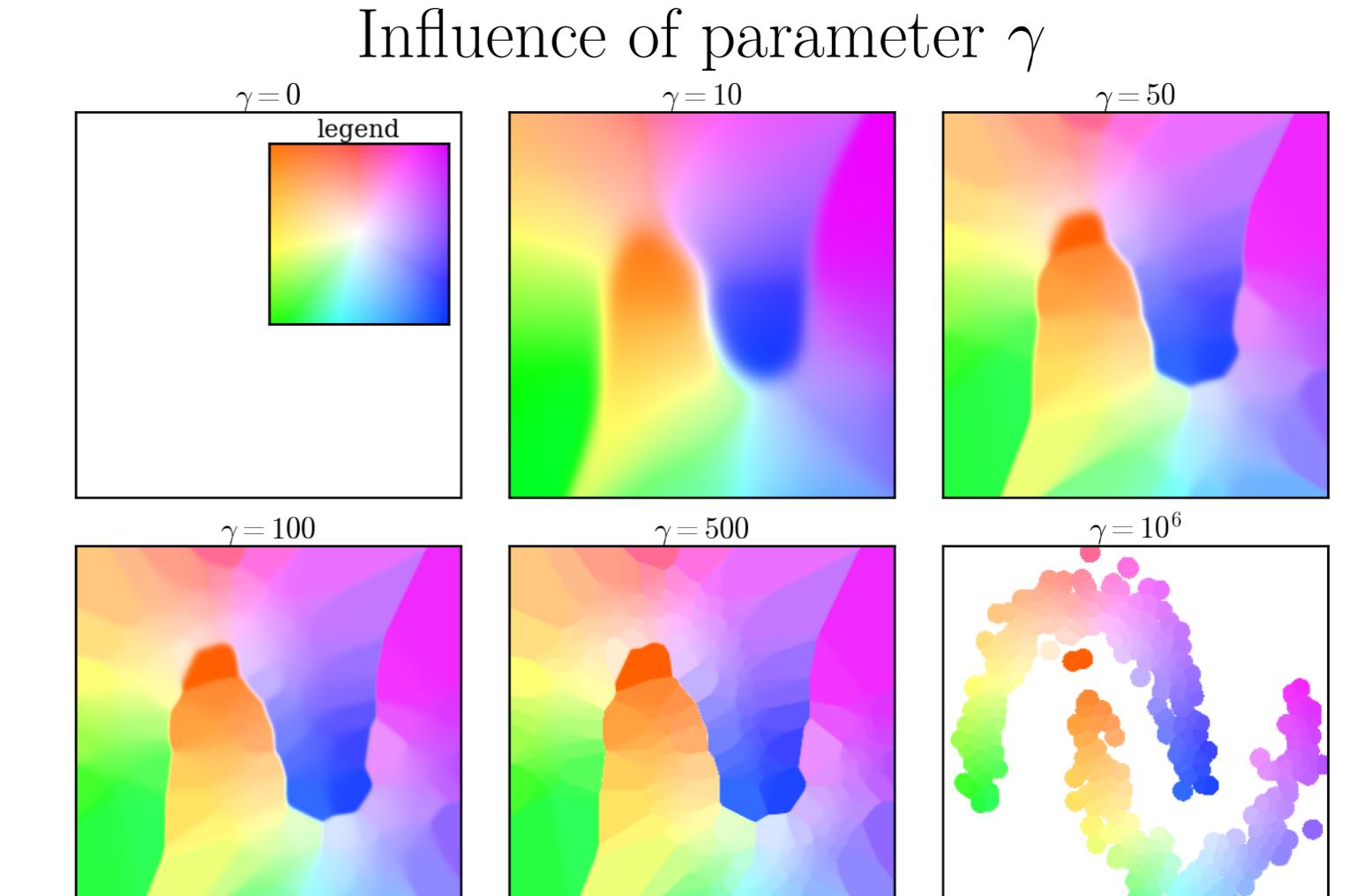


Leave-one-out cross-validation to search for optimal γ .

$$\hat{\gamma} = \arg \max_\gamma \frac{1}{n} \sum_{i=0}^n \mathbb{1}_{\{\hat{y}_i(\gamma) = y_i\}},$$

with

$$\hat{y}_i(\gamma) = \arg \min_y \|\varphi_\gamma^{\{-i\}}(\mathbf{x}_i) - \mathbf{t}_y\|^2$$



4 IMPLEMENTATION

Tricks

- 💡 Compute k -nn distances only

$$\tilde{\varphi}_\gamma(\mathbf{u}) = \mathbf{u} + \sum_{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{u})} w_\gamma(\mathbf{t}_{y_j} - \mathbf{x}_j)$$

where $\mathcal{N}_k(\mathbf{u})$ is the set of the k -nn of \mathbf{u} among the learning samples.

- 💡 Use Tree-based algorithms for k -nn search
⇒ Reduces computation complexity

- 💡 Keep only $n \times k$ distances in memory to re-compute the weights while varying γ for cross-validation.
⇒ Reduces memory storage requirement

Validation

Gaussian-SVM	k-nn	DT	KLIMT	KLIMT-SVM
$\gamma = 100$ $C = 10$	$k = 30$	min sample leaf = 40	$\gamma = 1500$	$\gamma = 1000$ $C = 0.1$
Grid search size 5 × 5	4	7	25	5 × 5
CV fitting time (s) 16 800	164	77	1.5	28 000
Prediction time (s) 0.4	0.1	0.01	0.1	0.7
ACC (%) 91	91	91	91	91

5 REAL DATA EXPERIMENT

- ZooScan data-set, **n=1.45 million** of plankton images (70% train, 30% test)
- **136 classes**
- **d = 10 features** from a pre-trained and fine-tuned ANN + PCA
- Huge **unbalance** between the classes: classical workaround (inverse class freq.)

	Random Forest	KLIMT
Cross-Validation	4-folds – 10 params.	leave-one-out – 10 params.
Fitting time (s)	3368	1143
Prediction time (s)	14	286
ACC (%)	73	71
Balanced ACC (%)	72	75

6 CONCLUSION

- ✓ Easy to tune (one parameter only)
- ✓ Interpretable → simple formulation
- ✓ Defines a kernel (e.g. SVM)
- ✓ Fast leave-one-out cross-validation, even with large data-sets
- ✓ Competitive results on a real data-set

7 FUTURE WORKS

- 💡 One parameter γ_i per learning sample
- 💡 Study the influence of the targets definition
- 💡 Regression on targets directly
- 💡 Approximate-nn search top speed up the prediction time