Anomaly Detection: class imbalance or novelty

Rémi Emonet

Université Jean-Monnet,
Laboratoire Hubert Curien,
Saint-Étienne

Talk at DÉFI IA 2020 (INSA Toulouse), 2020-01-23

Overview

- Introduction
- Anomaly and fraud detection
- Imbalanced classification problems
  - The Problem (and performance measures)
  - Reweight, resampling, etc
  - Learning maximum excluding ellipsoids
  - Correcting k-NN: γ-***NN***
  - Focusing on the F-Measure optimization
- Probabilistic models for unsupervised anomaly detection
- Discussion

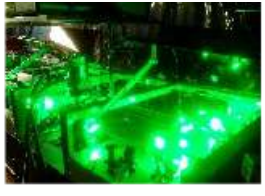
Laboratoire Hubert Curien

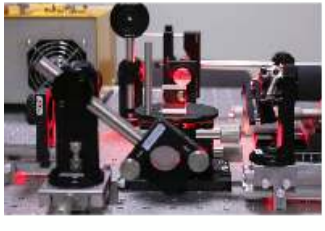
Joint Research Unit
of the University
&
CNRS (National Scientific Research Center)
&
Institute Optic Graduate School

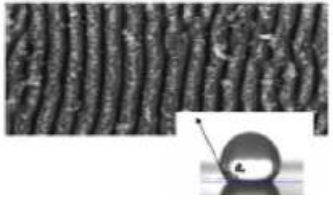
~ 230 Persons
(90 Researchers, 110 PhD students / postdoc)

Saint Etienne

- Optic-Photonic & Microwaves:
  - Micro & nano structuration
  - Laser Processes
  - Materials & Surfaces
  - Functionalization of surfaces
  - Materials for harsh environments
- Informatics, Telecom & Image:
  - Images Analysis
  - Data intelligence
  - Secured embedded Systems





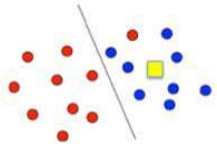


UNIVERSITÉ DE LYON
UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE
LABORATOIRE HUBERT CURIEN
UMR • CNRS • 5516 • SAINT-ETIENNE
INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech

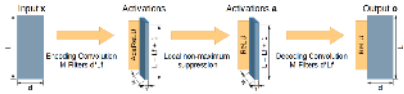
“Data Intelligence” Team:

Machine Learning and Complex Data Analysis

The team is specialized in statistical machine learning and data analysis and addresses mainly the following areas:

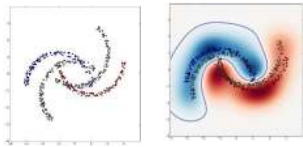


- **Representation Learning:** Deep Learning, Embedding for structured data (graphs, texts, images, sequences), incorporation of background knowledge and interpretability.



- **Metric Learning:** optimizing ad hoc metrics (distance/similarity) under semantic constraints.

- **Transfer Learning and Domain Adaptation:** Adapting and transferring models to new tasks or domains. Metric Learning

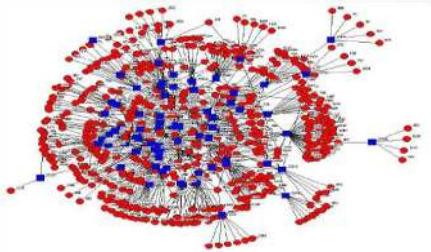


- **Learning theory:** Developing theoretical guarantees, formal frameworks and interpretation for learned models. (PAC-Bayesian Theory, Optimal Transport, ...)

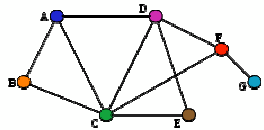
“Data Intelligence” Team:

Machine Learning and Complex Data Analysis

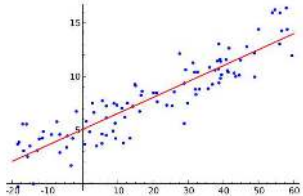
- **Data Mining:** Designing large scale methods to extract relevant and meaningful information from structured data, such as graphs or sequences, in the form of frequent or rare (spatio-temporal) patterns.

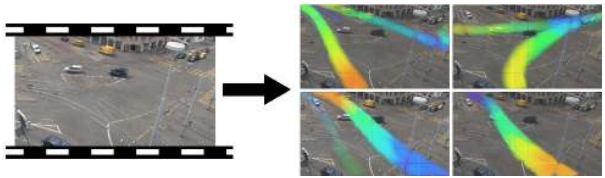


- **Learning/Analyzing from difficult scenarios:** Dealing with highly imbalanced data, few learning samples, incomplete data, privacy and fairness constraints.



- **Flagship Applications:**
  - Anomaly and Fraud Detection
  - Computer Vision
  - Medical data Analysis
  - Textual Data Analysis
  - Social Network Analy



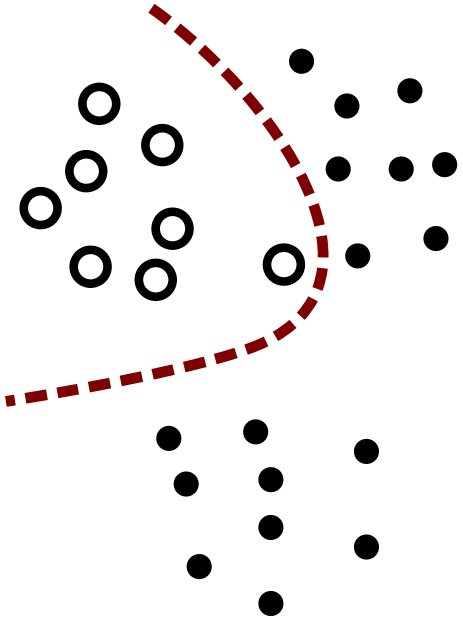


Overview

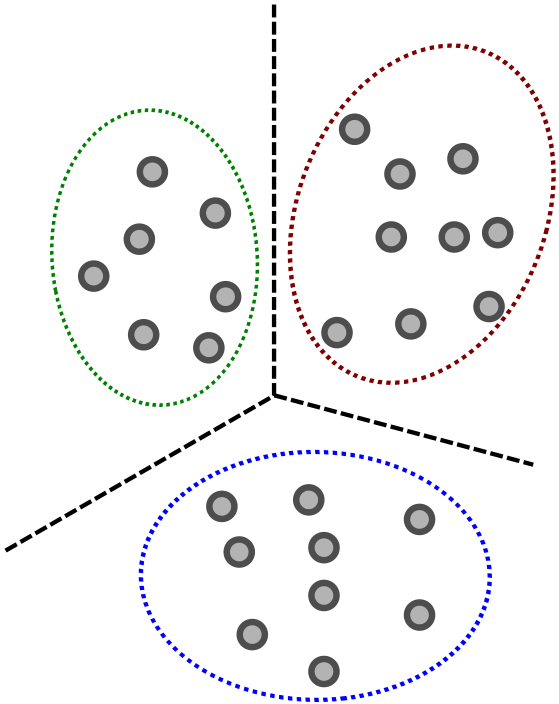
- Introduction
- Anomaly and fraud detection
- Imbalanced classification problems
  - The Problem (and performance measures)
  - Reweight, resampling, etc
  - Learning maximum excluding ellipsoids
  - Correcting k-NN: γ-***NN***
  - Focusing on the F-Measure optimization
- Probabilistic models for unsupervised anomaly detection
- Discussion

Supervised / Unsupervised *Machine Learning*

Supervised
(known labels)



Unsupervised
(only inputs)



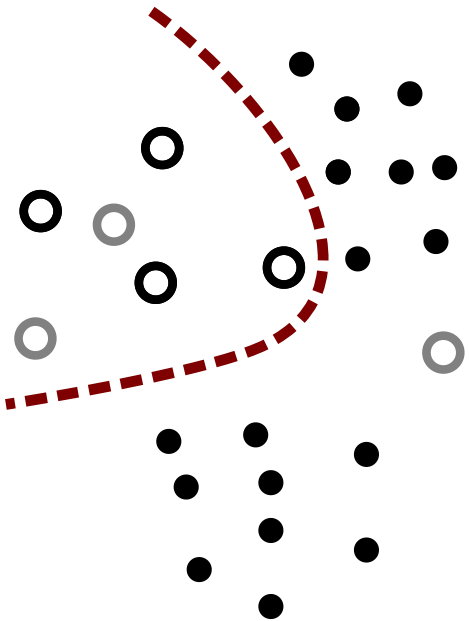
Supervised vs Unsupervised Learning

- Supervised
  - Given example inputs (X) and corresponding outputs (y)
  - Learn the fonction "input $\rightarrow$ output" ($y = f(X)$)
    - classification (categorical output)
    - regression (continous output)
  - methods:
    k-NN, SVM, SVR, Random Forests, Least Squares fit, Neural Networks, Gaussian Processes, Boosting, ...

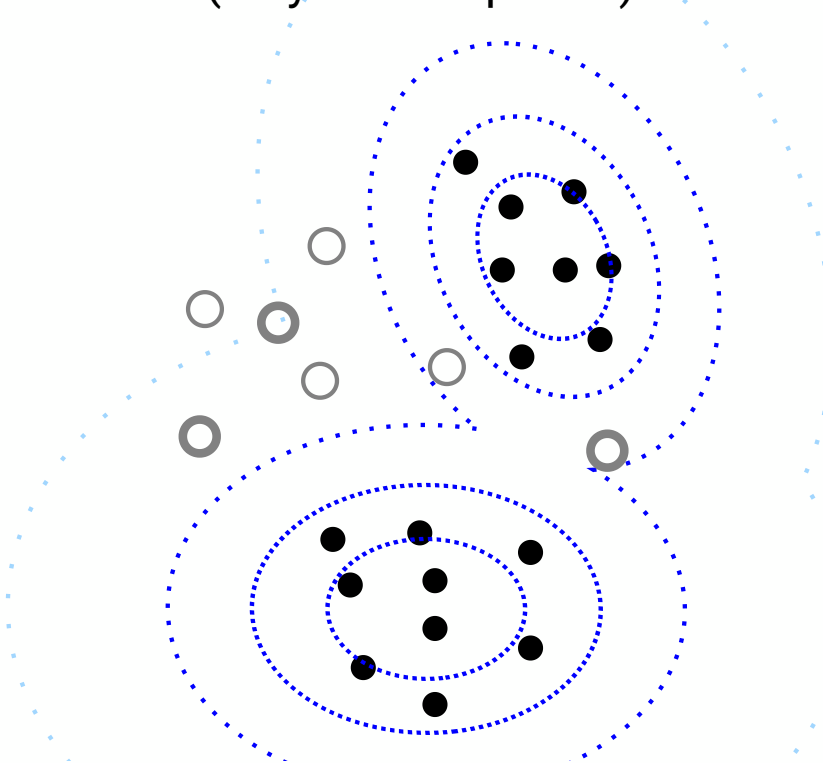
- Unsupervised
  - Given a set of data points (x)
  - Model/structure/understand this dataset
    - clustering, densitiy estimation
    - source separation
    - pattern and sequence mining
    - rare events / anomaly detection
  - Methods:
    PCA, k-means, OneClass-SVM, Isolation Forests, PGM (GMM, HMM, ...), DBSCAN, Autoencoders, GANs, KDE ...

Sup. / Unsup. *Anomaly Detection*

Supervised
(some anomalies)

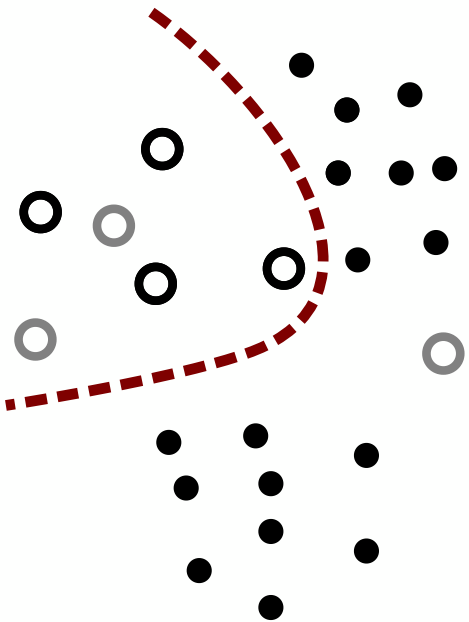


Unsupervised
(only normal points)

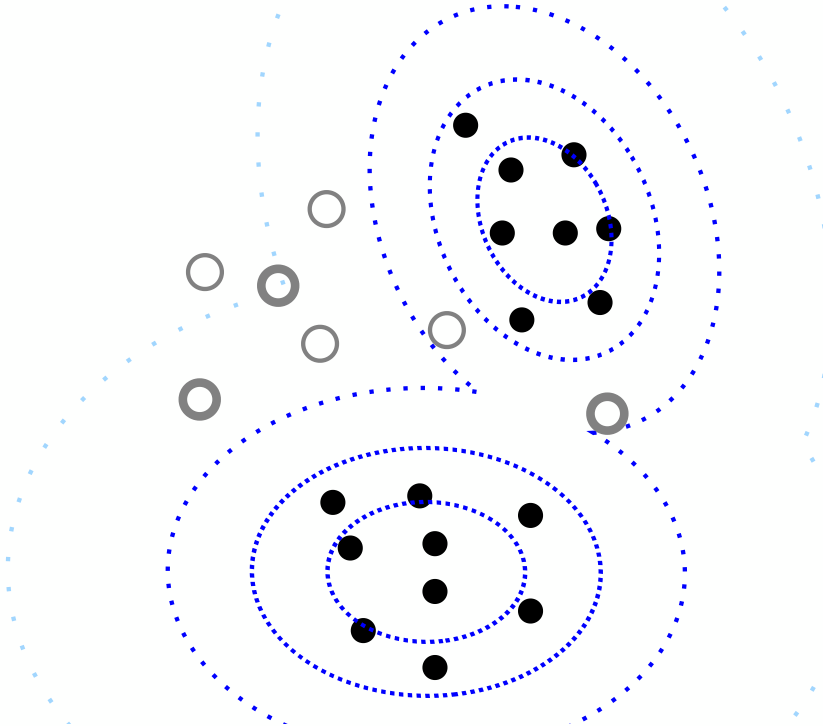


Class Imbalance or Novelty???

Supervised
(some anomalies)



Unsupervised
(only normal points)



Overview

- Introduction
- Anomaly and fraud detection
- Imbalanced classification problems
  - The Problem (and performance measures)
  - Reweight, resampling, etc
  - Learning maximum excluding ellipsoids
  - Correcting k-NN: γ-***NN***
  - Focusing on the F-Measure optimization
- Probabilistic models for unsupervised anomaly detection
- Discussion

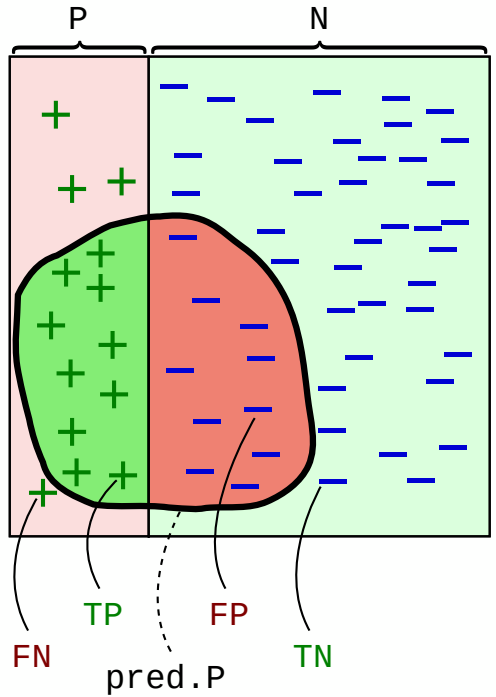
Imbalanced Problems: Examples

- Anomaly detection
  - unsafe situations in videos
  - defect detection in images
  - abnormal heart beat detection in ECG
- Fraud detection
  - fraudulent checks
  - credit card fraud (physical, online)
  - financial fraud (French DGFIP)

Imbalanced Classification Problems

- Binary classification
  - $+$ positive class: minority class, anomaly, rare event, …
  - $-$ negative class: majority class, normality, typical event, …
- Confusion matrix (of a model vs a ground truth)
  - TP: true positive
  - FP: false positive
  - TN: true negative
  - FN: false negative
- Some measures
  - Precision: $prec = \frac{TP}{TP + FP}$
  - Recall: $rec = \frac{TP}{P} = \frac{TP}{TP + FN}$
  - F_β-measure: $F_\beta = (1 + \beta^2) \frac{prec \cdot rec}{\beta^2 \cdot prec + rec}$

  (higher is better)



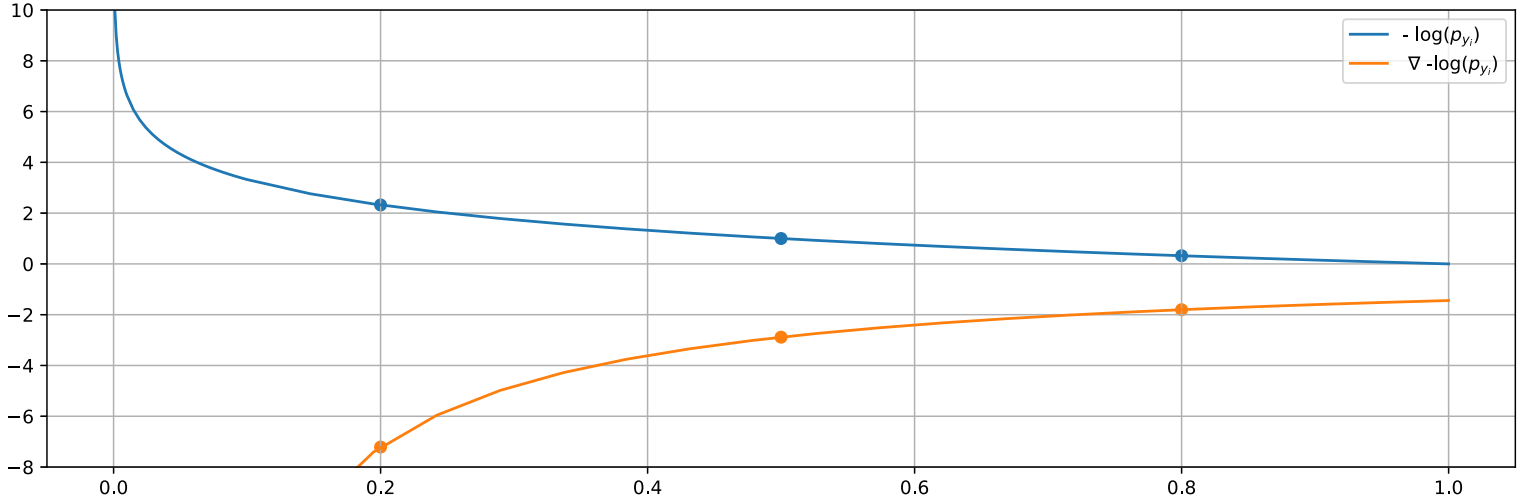
F-measure vs Accuracy ?

$$F_\beta = (1+\beta^2)\frac{prec \cdot rec}{\beta^2 \cdot prec + rec} = \frac{(1+\beta^2)\cdot(P-FN)}{1+\beta^2 P - FN + FP}$$

$$accuracy = \frac{TP+TN}{P+N} = 1 - \frac{FN+FP}{P+N}$$

- Accuracy inadequacy (e.g. $N = 10000, P = 100$)
  - lazy "all$-$" classifier ($TP = 0, TN = N, FP = 0, FN = P$)
  - $accuracy = \frac{0+N}{P+N} = \frac{10000}{10100} = 99\%$
  - $F_\beta = \frac{(1+\beta^2)(P-P)}{1+\beta^2 P - P + 0} = 0$
- F_β-measure challenges
  - discrete (like the accuracy)
  - non-convex (even with continuous surrogates)
  - **non-separable**, i.e. $F_\beta \neq \sum_{(x_i,y_i)\in S} \ldots$

Ok, but... I'm doing gradient descent, so...



- Gradient: $0.2 \Rightarrow -7.21$, $0.5 \Rightarrow -2.89$, $0.8 \Rightarrow -1.80$, $1 \Rightarrow -1.44$
- Example, gradient intensity is the same for:
  - $10+$ wrongly classified with an output proba. of 0.2
  - $40-$ correctly classified with an output proba 0.8
  - i.e., lazily predicting systematically 0.2 (for $+$) yields a "stable" solution with $10+$ vs $40-$

Ok, but… my deep model does 100%…

- … the 100% accuracy is on the train set
- … I cannot tell you if it will generalize well
- Our team is working on these aspects
  - APRIORI ANR project
  - guarantees for deep representation learning

Overview

- Introduction
- Anomaly and fraud detection
- Imbalanced classification problems
  - The Problem (and performance measures)
  - Reweight, resampling, etc
  - Learning maximum excluding ellipsoids
  - Correcting k-NN: γ-***NN***
  - Focusing on the F-Measure optimization
- Probabilistic models for unsupervised anomaly detection
- Discussion

Counteracting Imbalance

- Undersampling the majority class $-$
- Oversampling class $+$
- Generating fake $+$
- Using a weighted-classifiers learner

Overview

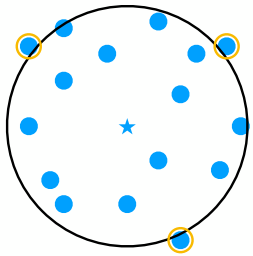
- Introduction
- Anomaly and fraud detection
- Imbalanced classification problems
  - The Problem (and performance measures)
  - Reweight, resampling, etc
  - Learning maximum excluding ellipsoids
  - Correcting k-NN: γ-***NN***
  - Focusing on the F-Measure optimization
- Probabilistic models for unsupervised anomaly detection
- Discussion

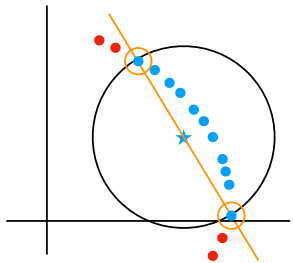
Learning maximum excluding ellipsoids from imbalanced data with theoretical guarantees

- **Guillaume Metzler**, Xavier Badiche, Brahim Belkasmi, Elisa Fromont, Amaury Habrard, Marc Sebban
- PRL2018 (Pattern Recognition Letters)
  .
- (slides borrowed from Guillaume Metzler Ph.D. defense)

Frauds are close to each other, they form small groups in the feature space

Given a set of m unlabelled points, find the center $\mathbf{c}$ and the **smallest** radius R of the ball that includes the data (Tax and Duin, 2004).





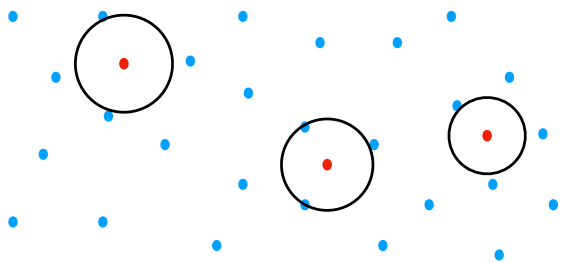
$$\begin{aligned} \min_{\mathbf{c},R,\xi} \quad & R^2 + \frac{\mu}{m}\sum_{i=1}^{m}\xi_i, \\ \text{s.t.} \quad & \|\mathbf{x}_i - \mathbf{c}\|_2^2 \leq R^2 + \xi_i, \ \forall i, \\ & \xi_i \geq 0 \ \forall i. \end{aligned}$$

$$\begin{aligned} \min_{\mathbf{c},\rho,\xi} \quad & \frac{1}{2}\|\mathbf{c}\|_2^2 + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i - \rho - \frac{1}{2}\|\mathbf{x}_i\|_2^2, \\ \text{s.t.} \quad & \mathbf{c}^T\mathbf{x}_i \geq \rho + \frac{1}{2}\|\mathbf{x}_i \ \forall i, \\ & \xi_i \geq 0 \ \forall i. \end{aligned}$$

Being in the ball $\iff$ being above the hyperplane

ME^2: Learning Risky Areas

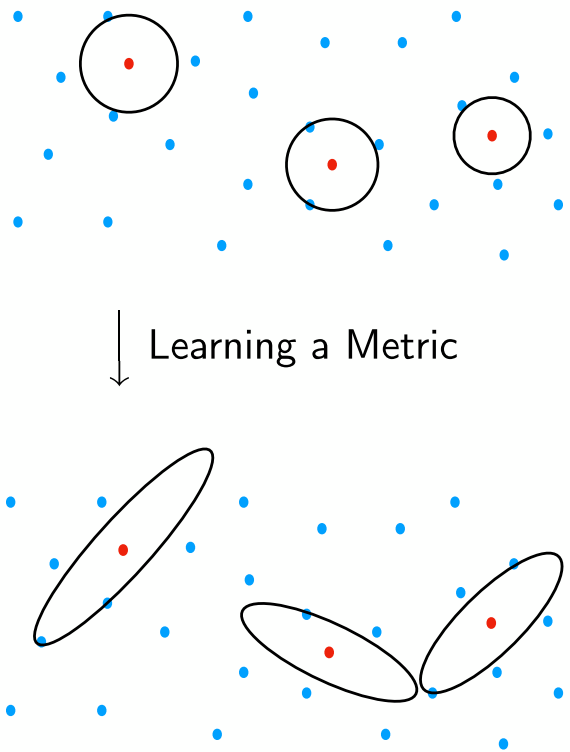
From MIB to ME^2



- Use the idea of MIB to create MEB
- One model per positive instance
- Require few positive neighbors

ME^2: Learning Risky Areas

From MIB to ME^2



- Use the idea of MIB to create MEB
- One model per positive instance
- Require few positive neighbors

- From balls to ellipsoids
- Increase decision boundary

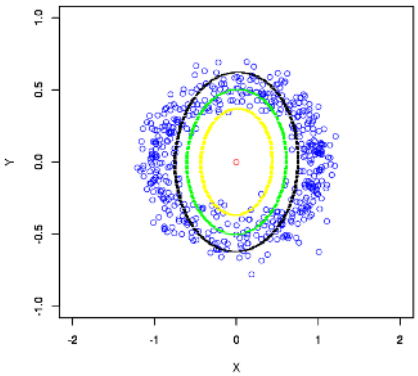
$\rightarrow$ Maximum Excluding Ellipsoids

ME^2: Learning Risky Areas

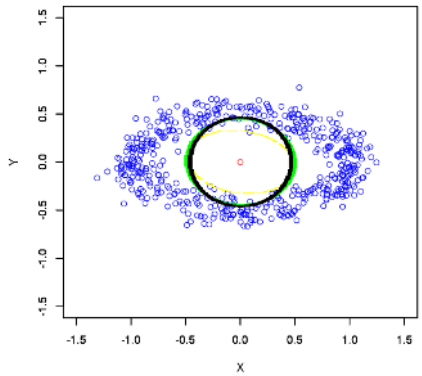
Optimization problem

$$\begin{aligned} \min_{R,\mathbf{M},\boldsymbol{\xi}} \quad & \frac{1}{m}\sum_{i=1}^{m}\xi_i + \mu(B-R)^2 + \lambda\|\mathbf{M}-\mathbf{I}\|_{\mathcal{F}^2}, \\ s.t. \quad & \|\mathbf{x}_i-\mathbf{c}\|_{\mathbf{M}}^2 \geq R-\xi_i, \quad \forall i=1,\dots,m, \\ & \xi_i \geq 0, \quad \forall i=1,\dots,m \\ & 0 \leq R \leq B, \end{aligned}$$

error terms (in terms of distances) regularization term



Influence of μ



Influence of λ

$\rightarrow$ express the Lagrangian $\mathcal{L}$ including the constraints
$\rightarrow$ expression of primal variables w.r.t. dual ones:

1. derivative of $\mathcal{L}$ w.r.t. primal variables
2. set derivatives to 0

$\rightarrow$ express the Lagrangian $\mathcal{L}$ including the constraints
$\rightarrow$ expression of primal variables w.r.t. dual ones:

1. derivative of $\mathcal{L}$ w.r.t. primal variables
2. set derivatives to 0

One of these derivatives gives:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{M}} = 0 \implies \mathbf{M} = \mathbf{I} + \frac{1}{2\lambda} \sum_{i=1}^{m} \alpha_k (\mathbf{x}_k - \mathbf{c})(\mathbf{x}_k - \mathbf{c})^T.$$

$\rightarrow$ $\mathbf{M}$ is Positive Semi Definite for free

Using stability framework (Bousquet and Elisseeff, 2002)

$$\mathcal{R}(\mathbf{M}, R) \leq \mathcal{R}_S(\mathbf{M}, R) + \mathcal{O}\left(\frac{1}{\min(\mu, \lambda)}\sqrt{\frac{\ln(1/\delta)}{2m}}\right),$$

where $\mathcal{R}_S(\mathbf{M}, R) = \frac{1}{m}\sum_{i=1}^{m}[R - \|\xi - \mathbf{c}\|^2_{\mathbf{M}}]_+$.

Using stability framework (Bousquet and Elisseeff, 2002)

$$\mathcal{R}(\mathbf{M}, R) \leq \mathcal{R}_S(\mathbf{M}, R) + \mathcal{O}\left(\frac{1}{\min(\mu, \lambda)}\sqrt{\frac{\ln(1/\delta)}{2m}}\right),$$

where $\mathcal{R}_S(\mathbf{M}, R) = \frac{1}{m}\sum_{i=1}^{m}[R - \|\xi - \mathbf{c}\|^2_{\mathbf{M}}]_+$.

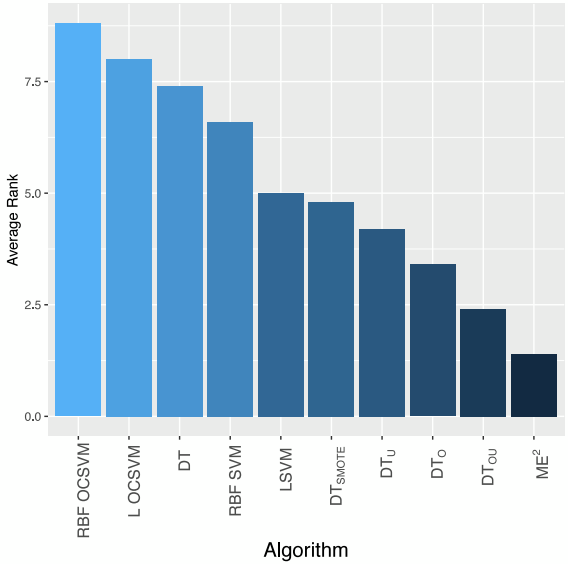
the true risk on the underlying and unknown distribution

the empirical risk over the sample S

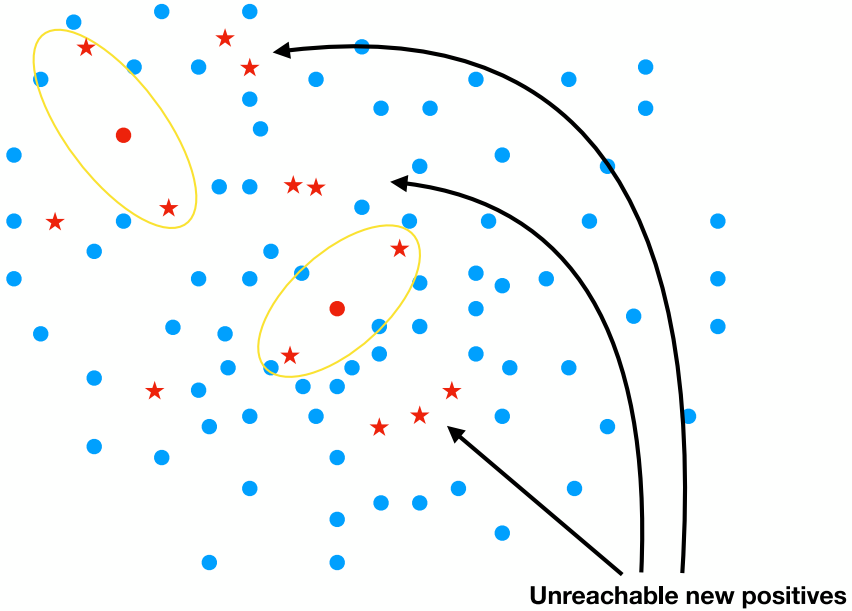
generalization gap of the learned model: depends on the complexity of the model

Comparison with standards algorithms on imbalanced datasets

Dataset	Nb. of ex.	% Pos.
Wine	1 599	3.3
Abalone17	2 338	2.5
Yeast6	1 484	2.4
Abalone20	1 916	1.4
Blitz	15 000	1.0



Lower Rank: able to reach better performance



Find a way to increase the influence zone of positives

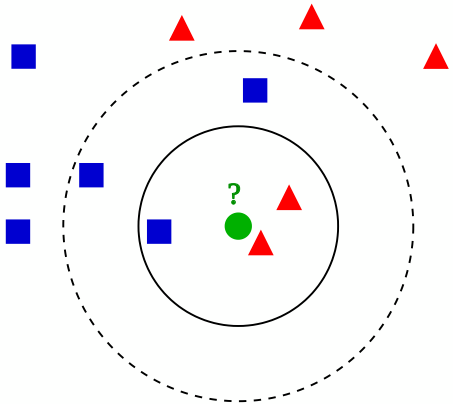
Overview

- Introduction
- Anomaly and fraud detection
- Imbalanced classification problems
  - The Problem (and performance measures)
  - Reweight, resampling, etc
  - Learning maximum excluding ellipsoids
  - Correcting k-NN: γ-***NN***
  - Focusing on the F-Measure optimization
- Probabilistic models for unsupervised anomaly detection
- Discussion

An Adjusted Nearest Neighbor Algorithm Maximizing the F-Measure from Imbalanced Data

- **Rémi Viola**, Rémi Emonet , Amaury Habrard, **Guillaume Metzler**, Sébastien Riou, Marc Sebban
- ICTAI2019

k-NN: k Nearest Neighbor Classification

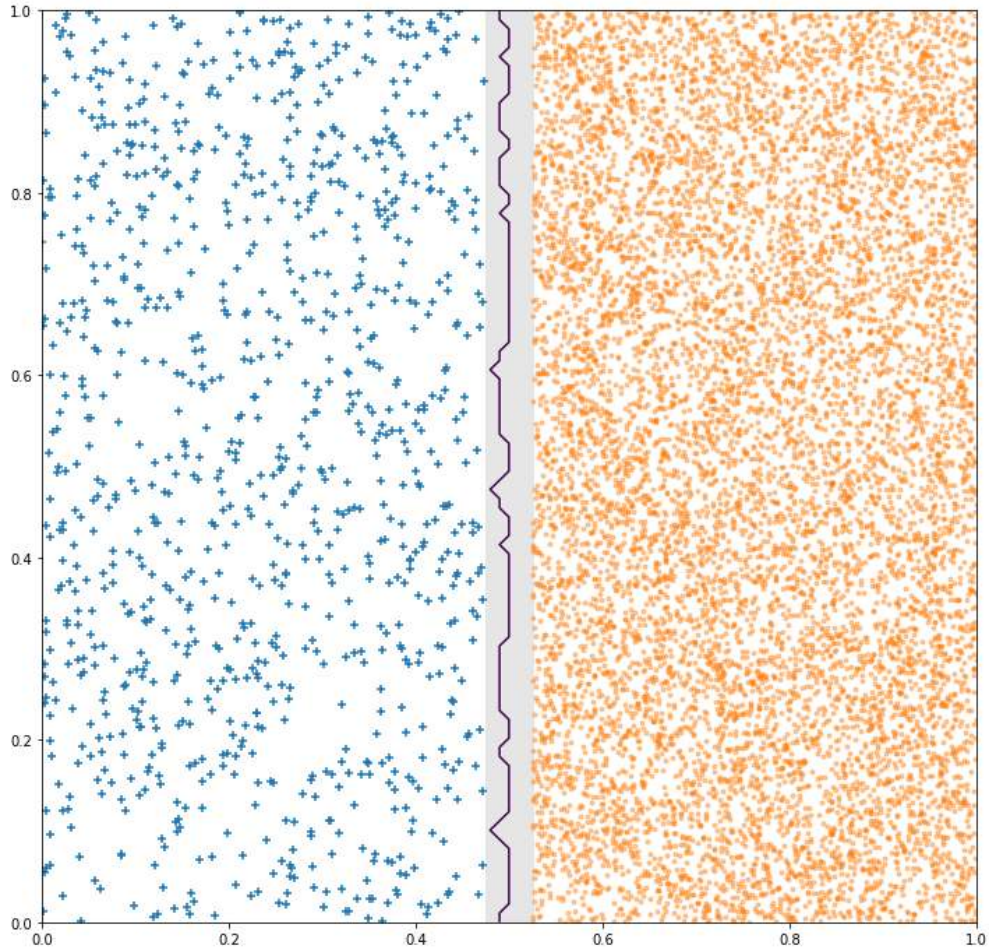


- k-NN
  - to classify a new point
  - find the closest k points (in the training section)
  - use a voting scheme to affect a class
  - efficient algorithms (K-D Tree, Ball Tree)
- Does k-NN still matter?
  - non-linear by design (with similarity to RBF-kernel SVM)
  - no learning, easy to patch a model (add/remove points)
  - Limits of k-NN for imbalanced data?

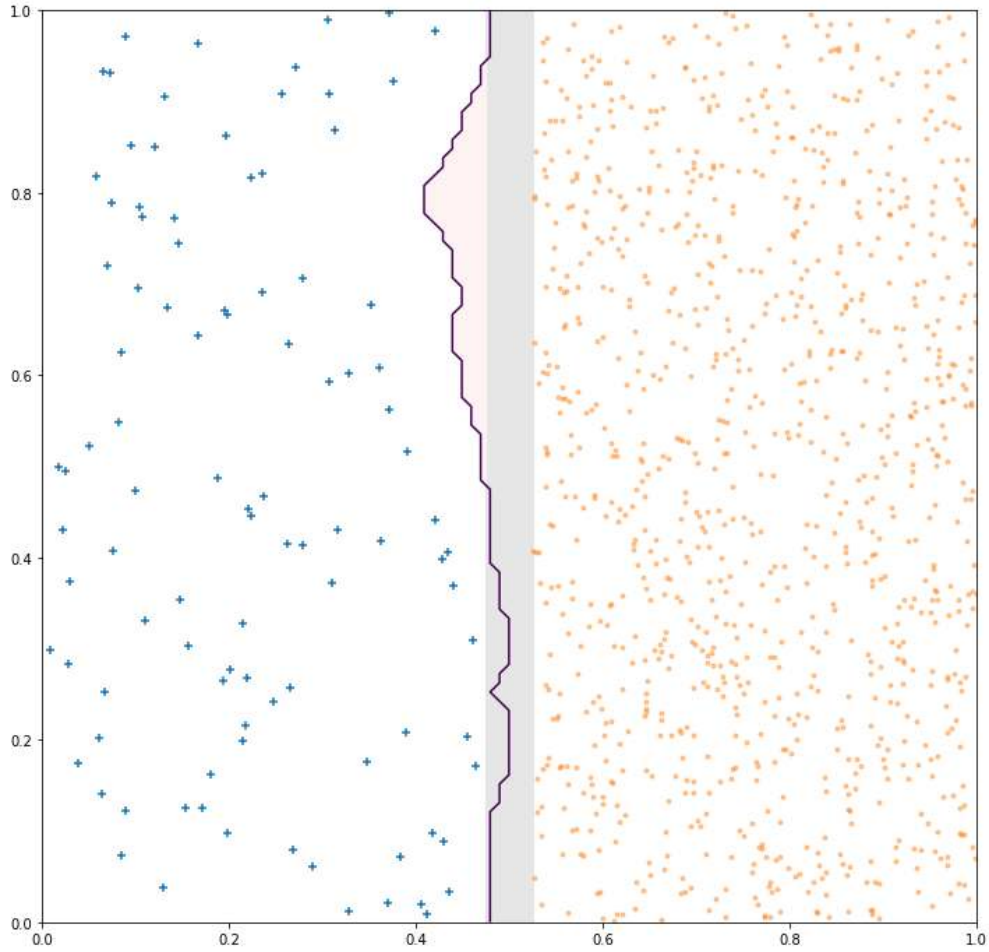
Limits of k-NN for imbalanced data?

1. k-NN behavior in uncertain areas
   - i.e., for some feature vector, the class can be $+$ or $-$
   - i.e., the Bayes Risk is non zero
   - ✔ not so bad, 1-NN respects imbalance (not k-NN)
2. k-NN behavior around boundaries
   - i.e., what happens if classes are separate but imbalanced
   - ✖ sampling effects cause problems

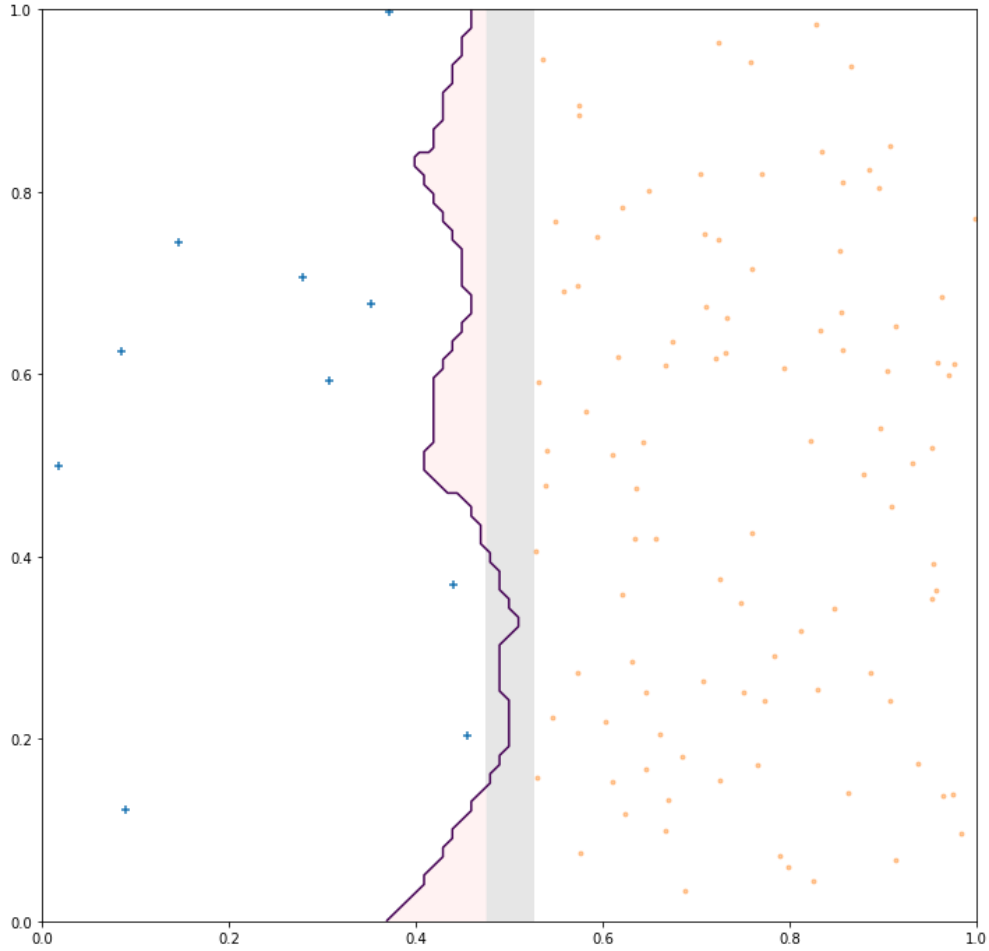
1-NN at a boundary (1000 + / 10k −)



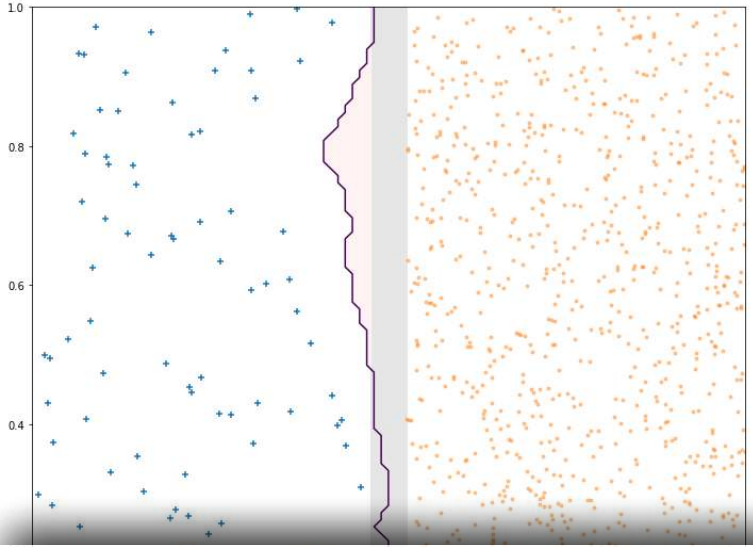
1-NN at a boundary (100 + / 1000 −)

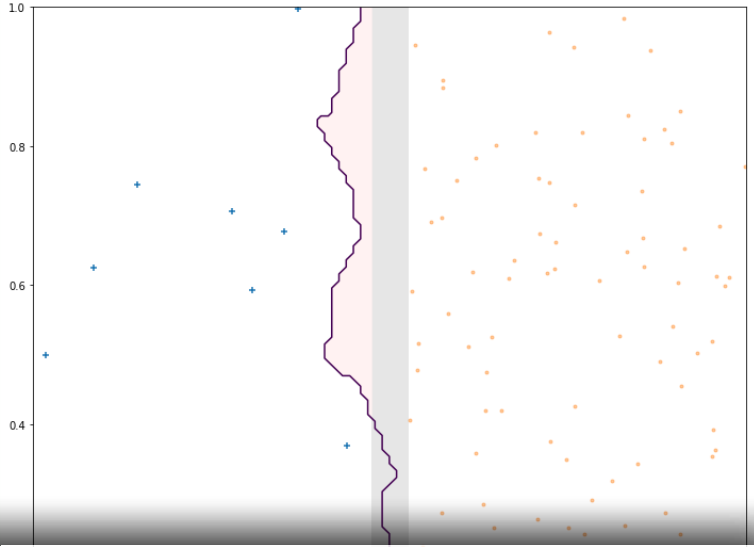


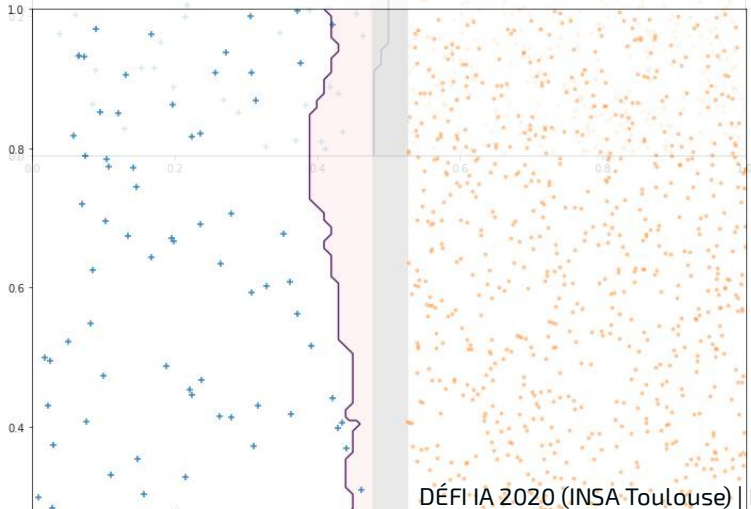
1-NN at a boundary (10 + / 100 −)

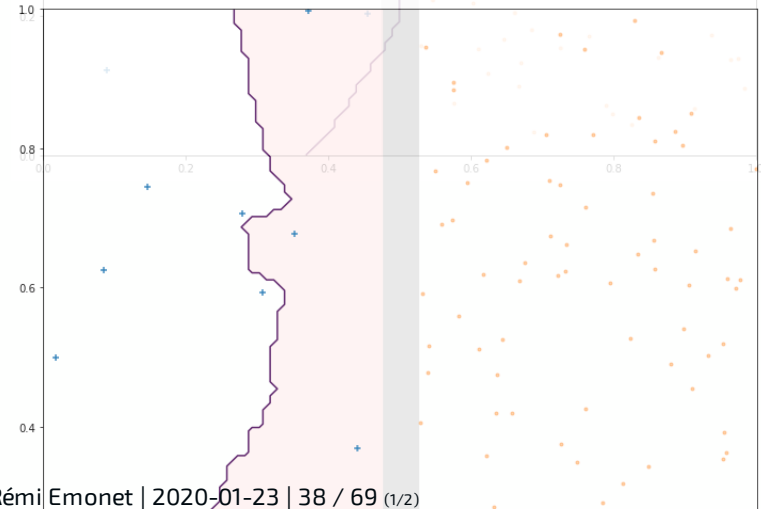


11-NN: increasing k?





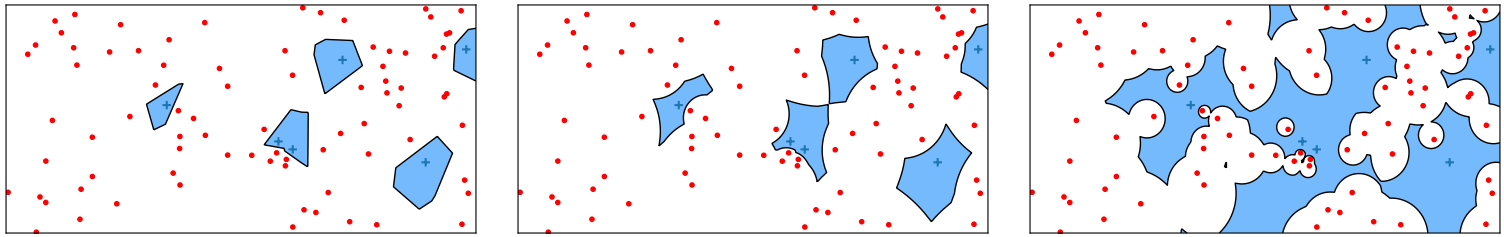




An Adjusted Nearest Neighbor Algorithm Maximizing the F-Measure from Imbalanced Data

- **Rémi Viola**, Rémi Emonet , Amaury Habrard, **Guillaume Metzler**, Sébastien Riou, Marc Sebban
- ICTAI2019

γ-NN Idea: push the decision boundary



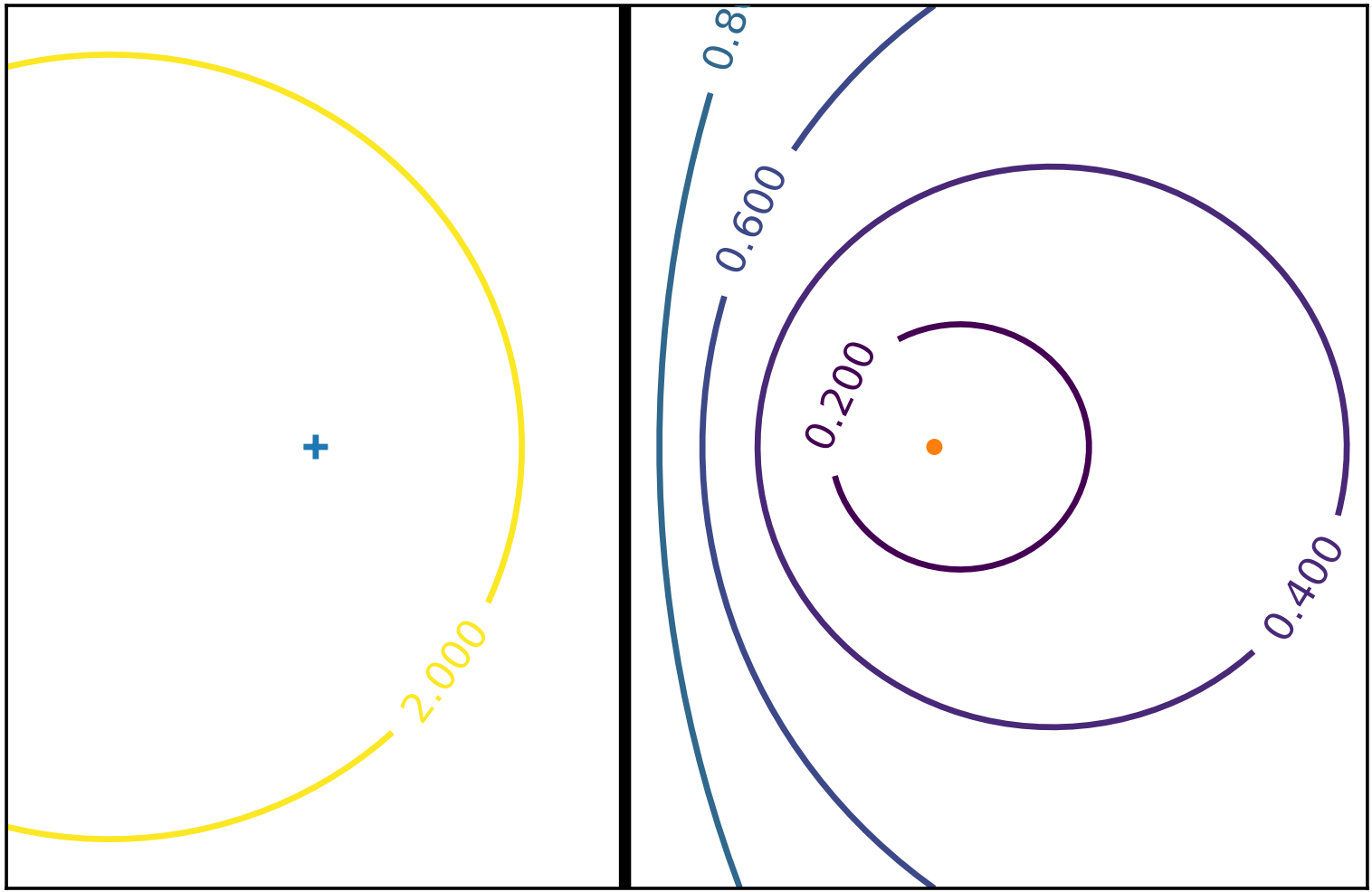
- Goal: correct for problems due to sampling with imbalance
- Genesis: GAN to generate "$+$" around existing ones

  $\Rightarrow$ unstable, failing, complex
- Approach
  - artificially make $+$ closer to new points
  - how? by using a different distance for $+$ and $-$
  - the base distance to $+$ gets multiplied by a parameter γ

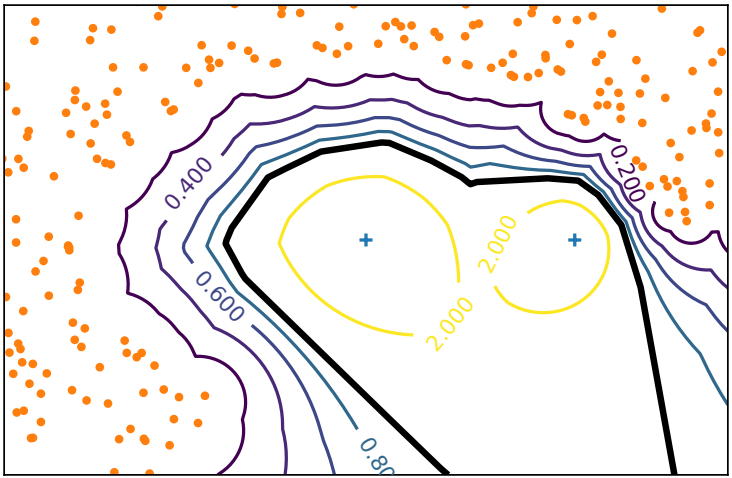
    (intuitively $\gamma \leq 1$ if $+$ is rare)

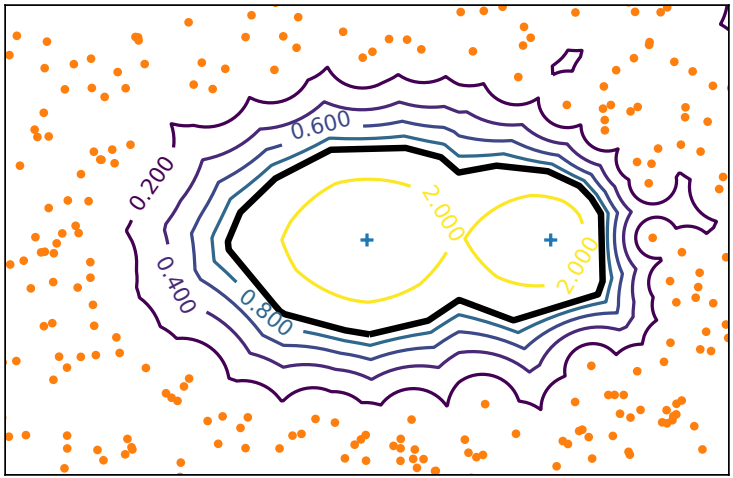
$$d_\gamma(x, x_i) = \begin{cases} d(x, x_i) & \text{if } x_i \in S_-, \\ \gamma \cdot d(x, x_i) & \text{if } x_i \in S_+. \end{cases}$$

γ-NN: varying γ with two points



γ-NN: varying γ with a few $+$





- γ-NN can control
  how close to the minuses it pushes the boundary

γ-NN: Algorithm

Algorithm 1: Classification of a new example with $\gamma k-$NN

Input : a query $\mathbf{x}$ to be classified, a set of labeled samples $S = S_+ \cup S_-$, a number of neighbors k, a positive real value γ, a distance function d

Output: the predicted label of $\mathbf{x}$

$\mathcal{NN}^-, \mathcal{D}^- \leftarrow nn(k, \mathbf{x}, S_-)$ // nearest negative neighbors with their distances

$\mathcal{NN}^+, \mathcal{D}^+ \leftarrow nn(k, \mathbf{x}, S_+)$ // nearest positive neighbors with their distances

$\mathcal{D}^+ \leftarrow \gamma \cdot \mathcal{D}^+$

$\mathcal{NN}_\gamma \leftarrow firstK\left(k, sortedMerge((\mathcal{NN}^-, \mathcal{D}^-), (\mathcal{NN}^+, \mathcal{D}^+))\right)$

$y \leftarrow +$ if $\left|\mathcal{NN}_\gamma \cap \mathcal{NN}^+\right| \geq \frac{k}{2}$ else $-$ // majority vote based on $\mathcal{NN}_\gamma$

return y

- Trivial to implement
- Same complexity as k-NN (at most twice)
- Training
  - none, as k-NN
  - γ is selected by cross-validation (on the measure of interest)

γ-NN: a way to reweight distributions

- In uncertain regions
- At the boundaries

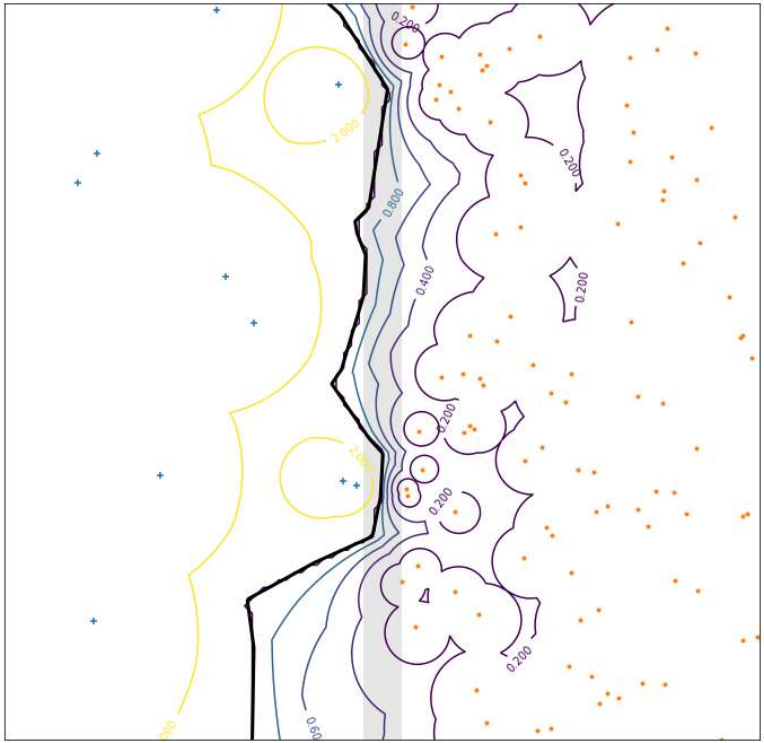
Results on public datasets (F-measure)

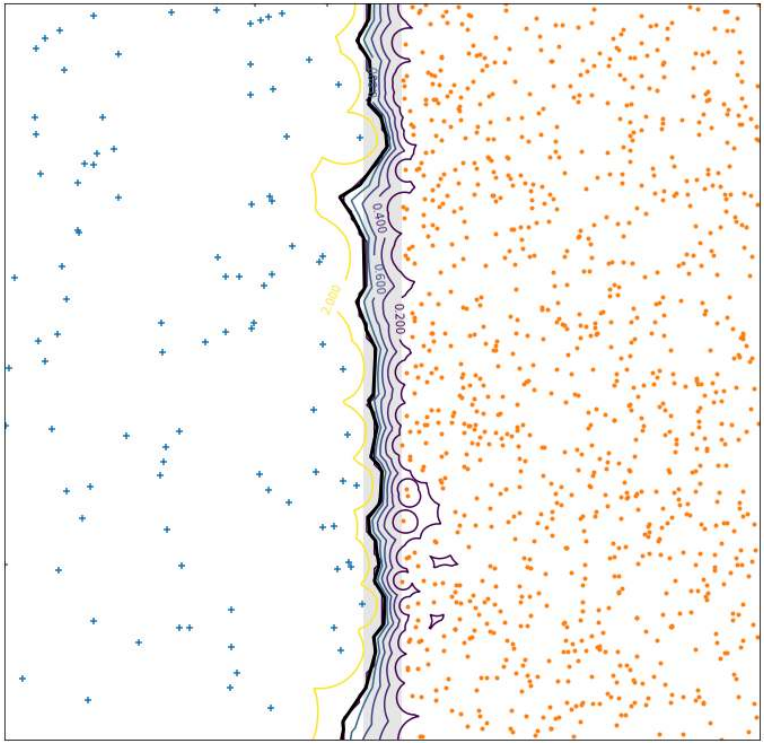
DATASETS	3−NN	DUPk−NN	Wk−NN	CWk−NN	LMNN	γk−NN
BALANCE	0.954(0.017)	0.954(0.017)	0.957(0.017)	0.961(0.010)	**0.963(0.012)**	0.954(0.029)
AUTOMPG	0.808(0.077)	0.826(0.033)	0.810(0.076)	0.815(0.053)	0.827(0.054)	**0.831(0.025)**
IONO	0.752(0.053)	0.859(0.021)	0.756(0.060)	0.799(0.036)	0.890(0.039)	**0.925(0.017)**
PIMA	0.500(0.056)	0.539(0.033)	0.479(0.044)	0.515(0.037)	0.499(0.070)	**0.560(0.024)**
WINE	0.881(0.072)	0.852(0.057)	0.881(0.072)	0.876(0.080)	**0.950(0.036)**	0.856(0.086)
GLASS	0.727(0.049)	0.733(0.061)	0.736(0.052)	0.717(0.055)	0.725(0.048)	**0.746(0.046)**
GERMAN	0.330(0.030)	0.449(0.037)	0.326(0.030)	0.344(0.029)	0.323(0.054)	**0.464(0.029)**
VEHICLE	0.891(0.044)	0.867(0.027)	0.891(0.044)	0.881(0.021)	**0.958(0.020)**	0.880(0.049)
HAYES	0.036(0.081)	0.183(0.130)	0.050(0.112)	0.221(0.133)	0.036(0.081)	**0.593(0.072)**
SEGMENTATION	0.859(0.028)	0.862(0.018)	0.877(0.028)	0.851(0.022)	**0.885(0.034)**	0.848(0.025)
ABALONE8	0.243(0.037)	0.318(0.013)	0.241(0.034)	0.330(0.015)	0.246(0.065)	**0.349(0.018)**
YEAST3	0.634(0.066)	0.670(0.034)	0.634(0.066)	**0.699(0.015)**	0.667(0.055)	0.687(0.033)
PAGEBLOCKS	0.842(0.020)	0.850(0.024)	0.849(0.019)	0.847(0.029)	**0.856(0.032)**	0.844(0.023)
SATIMAGE	0.454(0.039)	0.457(0.027)	0.454(0.039)	0.457(0.023)	**0.487(0.026)**	0.430(0.008)
LIBRAS	**0.806(0.076)**	0.788(0.187)	**0.806(0.076)**	0.789(0.097)	0.770(0.027)	0.768(0.106)
WINE4	0.031(0.069)	**0.090(0.086)**	0.031(0.069)	0.019(0.042)	0.000(0.000)	**0.090(0.036)**
YEAST6	0.503(0.302)	0.449(0.112)	0.502(0.297)	0.338(0.071)	0.505(0.231)	**0.553(0.215)**
ABALONE17	0.057(0.078)	**0.172(0.086)**	0.057(0.078)	0.096(0.059)	0.000(0.000)	0.100(0.038)
ABALONE20	0.000(0.000)	0.000(0.000)	0.000(0.000)	**0.067(0.038)**	0.057(0.128)	0.052(0.047)
MEAN	0.543(0.063)	0.575(0.053)	0.544(0.064)	0.559(0.046)	0.560(0.053)	**0.607(0.049)**

Results on DGFiP datasets (F-measure)

DATASETS	3−NN	γk−NN	SMOTE	SMOTE+γk−NN
DGFIP19 2	0,454 (0,007)	0,528 (0,005)	0,505 (0,010)	**0,529** (0,003)
DGFIP9 2	0,173 (0,074)	0,396 (0,018)	0,340 (0,033)	**0,419** (0,029)
DGFIP4 2	0,164 (0,155)	0,373 (0,018)	0,368 (0,057)	**0,377** (0,018)
DGFIP8 1	0,100 (0,045)	**0,299** (0,010)	0,278 (0,043)	**0,299** (0,011)
DGFIP8 2	0,140 (0,078)	0,292 (0,028)	**0,313** (0,048)	0,312 (0,021)
DGFIP9 1	0,088 (0,090)	0,258 (0,036)	0,270 (0,079)	**0,288** (0,026)
DGFIP4 1	0,073 (0,101)	0,231 (0,139)	0,199 (0,129)	**0,278** (0,067)
DGFIP16 1	0,049 (0,074)	0,166 (0,065)	0,180 (0,061)	**0,191** (0,081)
DGFIP16 2	0,210 (0,102)	0,202 (0,056)	0,220 (0,043)	**0,229** (0,026)
DGFIP20 3	0,142 (0,015)	0,210 (0,019)	0,199 (0,015)	**0,212** (0,019)
DGFIP5 3	0,030 (0,012)	0,105 (0,008)	**0,110** (0,109)	0,107 (0,010)
MEAN	0,148 (0,068)	0,278 (0,037)	0,271 (0,057)	**0,295** (0,028)

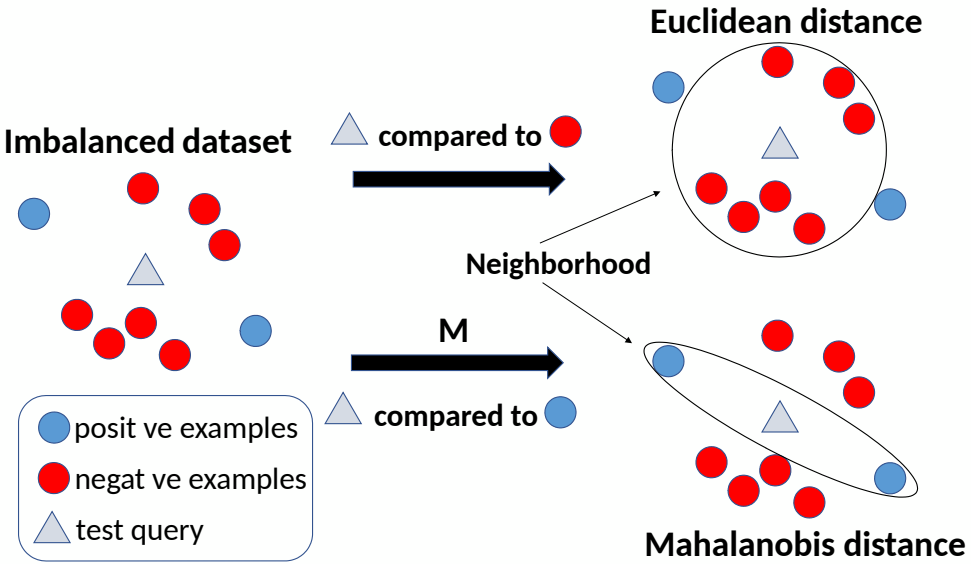
γ-NN at a boundary (10 and 100 $+$)





(some) Work in progress

- Note:
  - γ-NN learns a metric for comparing a query to a $+$
  - γ-NN kind of learn the size of a sphere around $+$
  - this is “Metric Learning”
- Extension
  - learn a full metric (a matrix M and not only γ)
  - derive a learning algorithm (not just cross-validation)



$$\min_{\mathbf{M}\in\mathbb{S}^+} \frac{1}{m^3}\left((1-\alpha)\sum_{\substack{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k)\\ y_i=y_j=1\neq y_k}} \ell_{\text{FN}}(\mathbf{M},\mathbf{z}_i,\mathbf{z}_j,\mathbf{z}_k) \;+\; \alpha \sum_{\substack{(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k)\\ y_i=y_j=-1\neq y_k}} \ell_{\text{FP}}(\mathbf{M},\mathbf{z}_i,\mathbf{z}_j,\mathbf{z}_k)\right) + \mu\|\mathbf{M}-\mathbf{I}\|_{\mathcal{F}}^2,$$

such that $\lambda_{\max}(\mathbf{M}) \leq 1$.

where $\mathbb{S}^+$ is the set of PSD matrices, $\lambda_{\max}(\mathbf{M})$ is the largest eigenvalue of the PSD matrix $\mathbf{M}$, ℓ_{FN} and ℓ_{FP} are defined by:

$$\ell_{\text{FN}}(\mathbf{M},\mathbf{z}_i,\mathbf{z}_j,\mathbf{z}_k) = [1-c+d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j)^2 - d(\mathbf{x}_i,\mathbf{x}_k)^2]_+,$$
$$\ell_{\text{FP}}(\mathbf{M},\mathbf{z}_i,\mathbf{z}_j,\mathbf{z}_k) = [1-c+d(\mathbf{x}_i,\mathbf{x}_j)^2 - d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k)^2]_+,$$

Overview

- Introduction
- Anomaly and fraud detection
- Imbalanced classification problems
  - The Problem (and performance measures)
  - Reweight, resampling, etc
  - Learning maximum excluding ellipsoids
  - Correcting k-NN: γ-***NN***
  - Focusing on the F-Measure optimization
- Probabilistic models for unsupervised anomaly detection
- Discussion

From Cost-Sensitive Classification to Tight F-measure Bounds

- **Kevin Bascol**, Rémi Emonet, Elisa Fromont, Amaury Habrard, **Guillaume Metzler**, Marc Sebban
- AISTATS2019

Optimizing the F_β-measure?

- Reminder
  - Precision: $prec = \frac{TP}{TP + FP}$
  - Recall: $rec = \frac{TP}{P} = \frac{TP}{TP + FN}$
  - F_β-measure: $F_\beta = (1 + \beta^2) \frac{prec \cdot rec}{\beta^2 \cdot prec + rec}$
- **Non-separability**, i.e. $F_\beta \neq \sum_{(x_i, y_i) \in S} \dots$

NB: accuracy is separable, $acc = \sum_{(x_i, y_i) \in S} \frac{1}{m} \delta(y_i - \hat{y}_i)$

⇒ The loss for one point depends on the others
⇒ Impossible to optimize directly
⇒ Impossible to optimize on a subset (minibatch)

Weighted classification for F_β

$$F_\beta = \frac{(1+\beta^2)\cdot(P-FN)}{1+\beta^2 P - FN + FP} = \frac{(1+\beta^2)\cdot(P-e_1)}{1+\beta^2 P - e_1 + e_2}$$

- The F_β-measure is linear fractional *(in $e = (e_1, e_2) = (FN, FP)$)*
  i.e. $F_\beta = \dfrac{\langle a', e\rangle + b}{\langle c, e\rangle + d} = \dfrac{A}{B}$
- Relation to weighted classification

$$\begin{aligned}
& F_\beta \geq t \qquad \text{(we achieve a good, above } t\text{, } F_\beta \text{ value)} \\
\Leftrightarrow\ & A \geq t \cdot B \\
\Leftrightarrow\ & A - t\cdot B \geq 0 \\
\Leftrightarrow\ & (1+\beta^2)\cdot(P-e_1) - t(1+\beta^2 P - e_1 + e_2) \geq 0 \\
\Leftrightarrow\ & (-1-\beta^2+t)e_1 - te_2 \geq -P(1+\beta^2) + t(1+\beta^2 P) \\
\Leftrightarrow\ & (1+\beta^2-t)e_1 + te_2 \leq -P(1+\beta^2) + t(1+\beta^2 P)
\end{aligned}$$

⇒ so, we can minimize the weighted problem
with class weights $a(t) = (1+\beta^2 - t, t)$

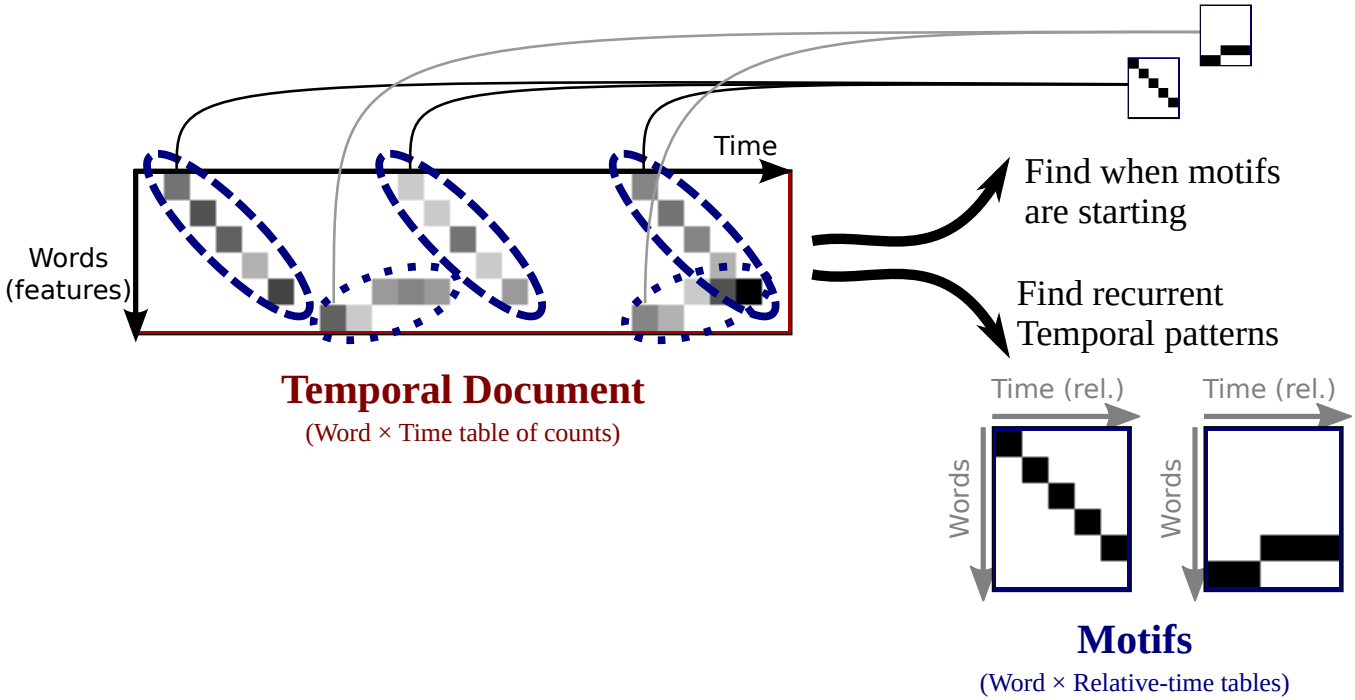
CONE Demo...

Overview

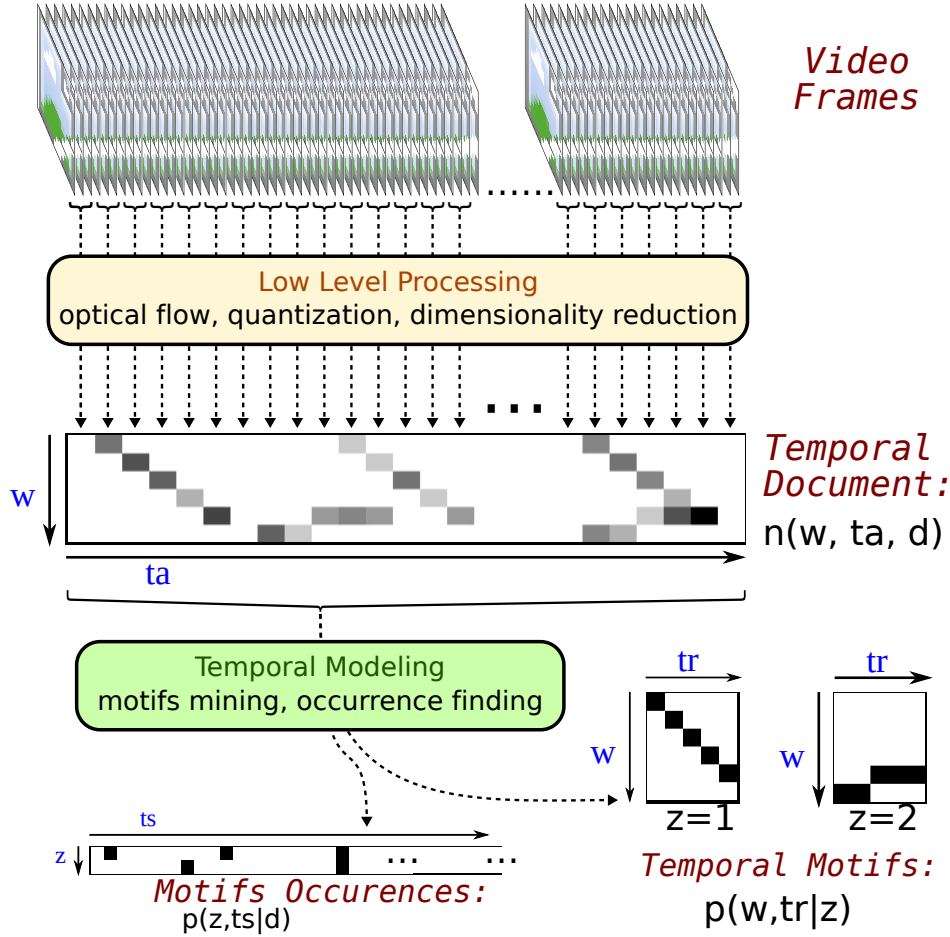
- Introduction
- Anomaly and fraud detection
- Imbalanced classification problems
  - The Problem (and performance measures)
  - Reweight, resampling, etc
  - Learning maximum excluding ellipsoids
  - Correcting k-NN: γ-***NN***
  - Focusing on the F-Measure optimization
- Probabilistic models for unsupervised anomaly detection
- Discussion

Learning normality

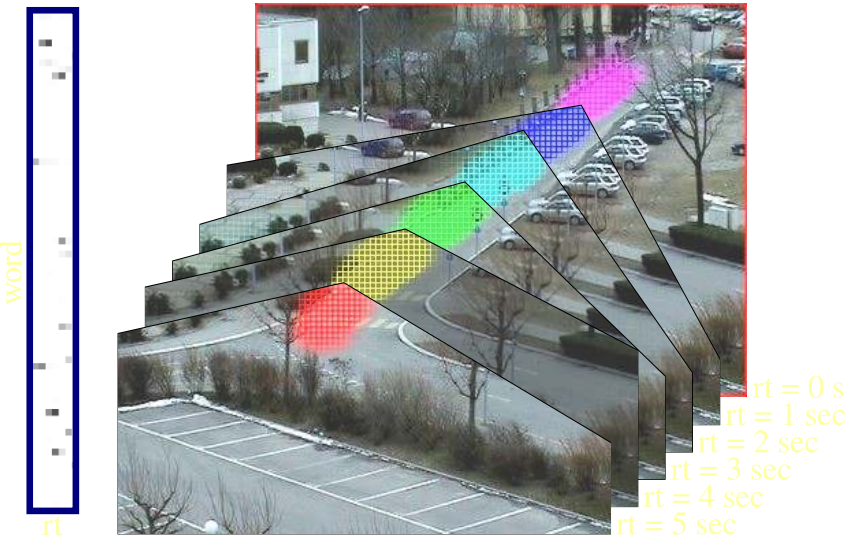
Unsupervised Temporal Motif Mining in videos / temporal data (spectrograms, ...)

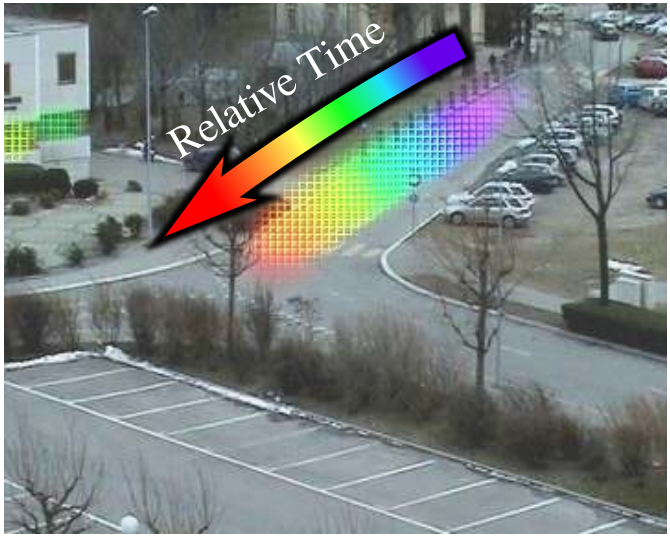


Temporal Patterns in Videos: Full Process

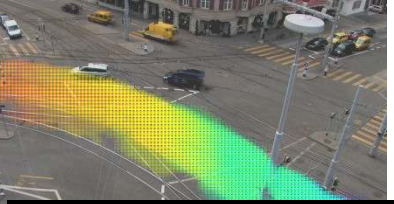


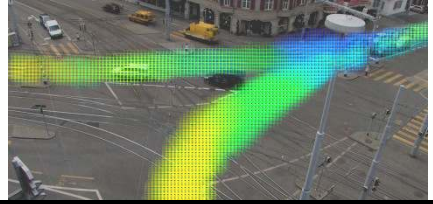
Video Motif Representation



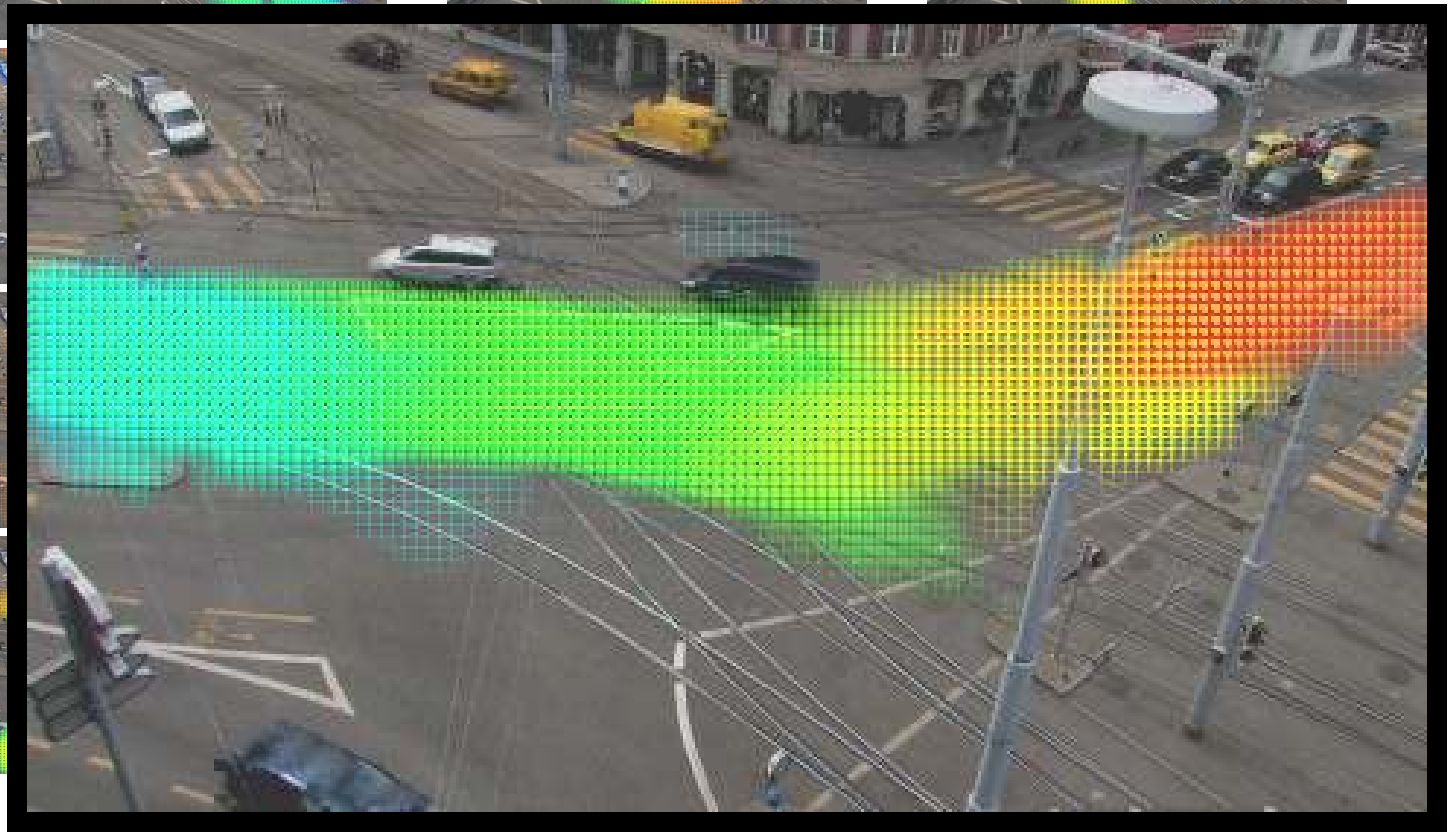


Example Motifs Obtained from a Static Camera

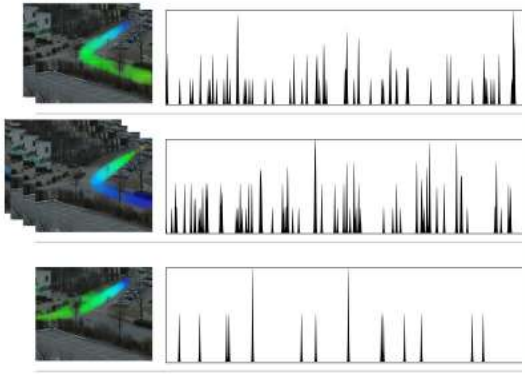




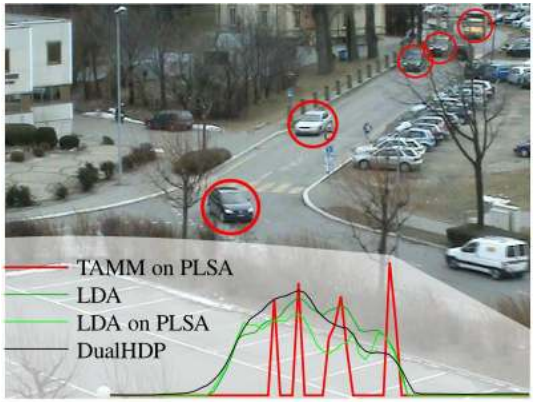




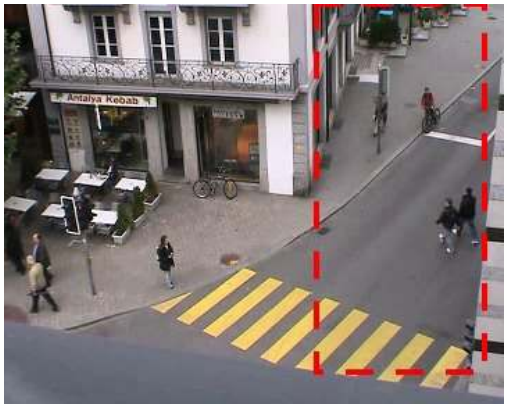
Application with Static Cameras



scene understanding



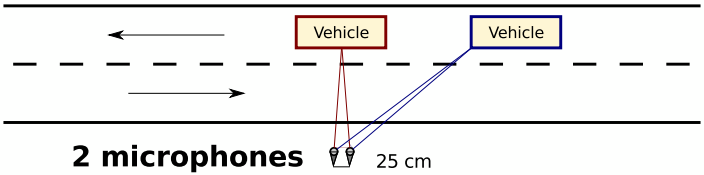
car counting



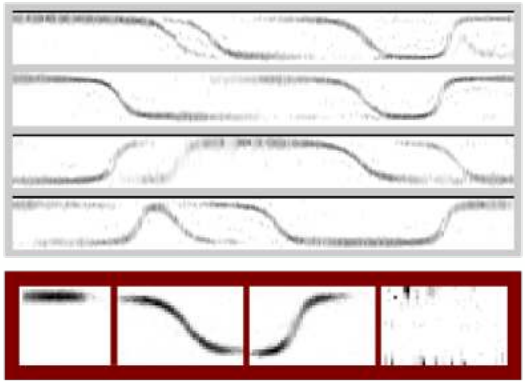
anomaly detection

stream selection, anomaly detection, multi-camera analysis

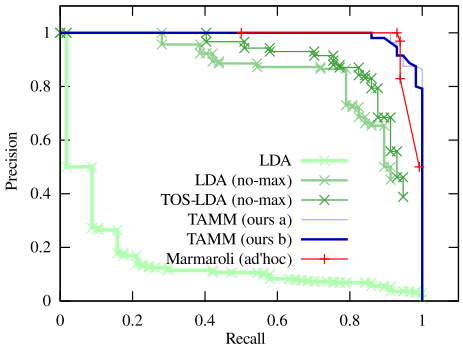
Audio data?



a pair of microphones…



… meaningful motifs…



… and good counting results

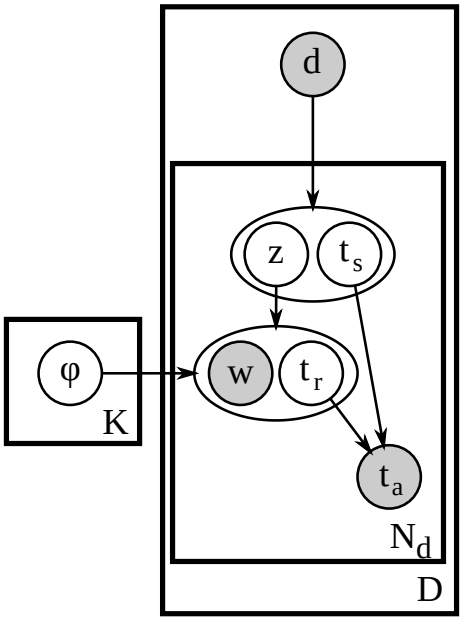
… also with spectrograms

How Can We Do This?!

Sol. 1: Hierarchical Probabilistic Models

$$\mathcal{L} = \sum_{d} \sum_{w} \sum_{t_a} n(w, t_a, d) log \sum_{z} \sum_{t_s} p(w, t_r|z) p(z, t_s|d)$$

- Generative Model
  ⇒ interpretable by design
- ⚠ unknown number of motifs
  ⇒ use infinite models
- Inference
  - maximum likelihood, EM like
- Sparsity on occurrences $p(ts|z, d)$
  - new objective function:
    $\mathcal{L} - \lambda_{sparse} \sum_{d} \sum_{z} KL(U||p(ts|z, d))$

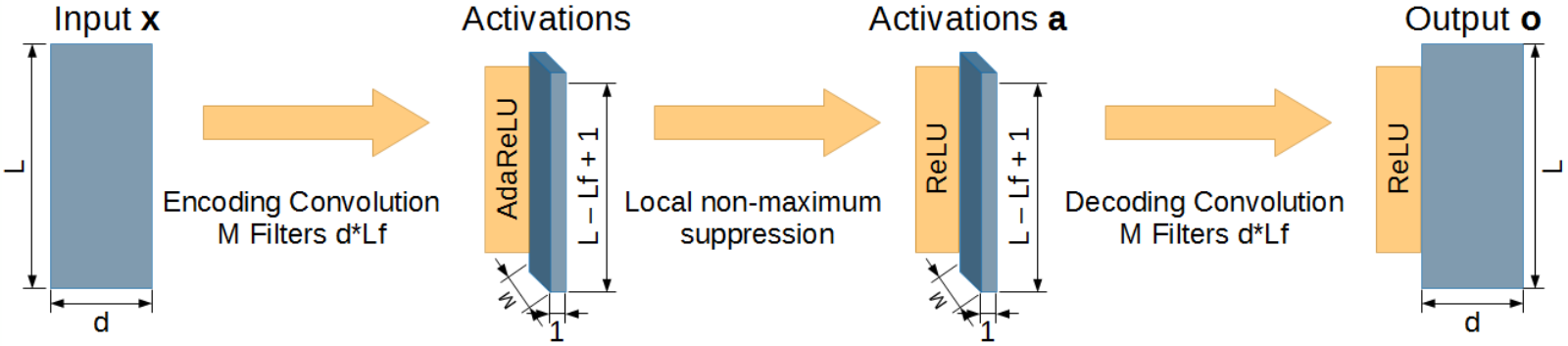


Sol. 2: Neural Networks, Auto-encoders

- Principle of auto-encoders
  - learn to produce the input from the input
  - going through a limited-size representation (bottleneck)
  - $x' = f(x) = f_{DEC}(f_{ENC}(x))$
  - minimize the reconstruction error $d(x, x') = \|x - x'\|^2$

- ⚠ Issues: interpretability, number of motifs, ...

Sol. 2: Interpretable Auto-Encoders



- Add specific operators (layers)
  - global specialized maximum selection (AdaReLU)
  - locally, filter response decorrelation
- Special Loss: a combination of well-chosen target functions
  - encourage sparse motifs (with a lot of zero)
  - encourage sparse activations
  - ⚠ unknown number of motifs ⇒ use “group-sparsity”

Overview

- Introduction
- Anomaly and fraud detection
- Imbalanced classification problems
  - The Problem (and performance measures)
  - Reweight, resampling, etc
  - Learning maximum excluding ellipsoids
  - Correcting k-NN: γ-***NN***
  - Focusing on the F-Measure optimization
- Probabilistic models for unsupervised anomaly detection
- Discussion

Thank you! Questions?