One-class classification pour la détection de surface

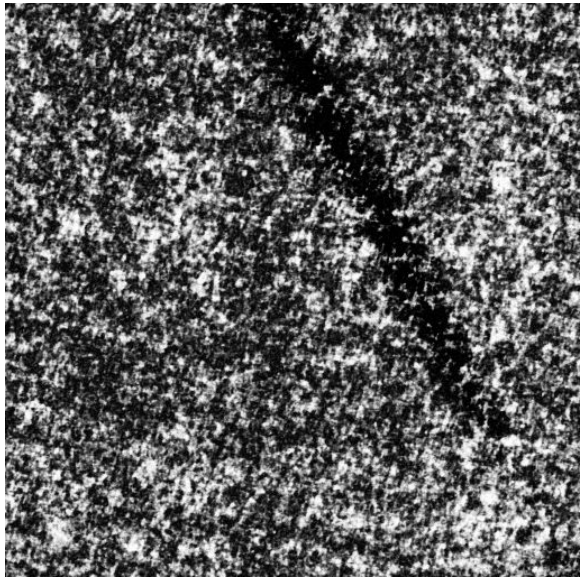
Étudiants
Alexandre MILESI
Sylvain MARCHIENNE

Superviseurs
Jonathan DEKHTIAR
Alexandre DURUPT

Mardi 3 Juillet 2018

One-class classification et DAGM

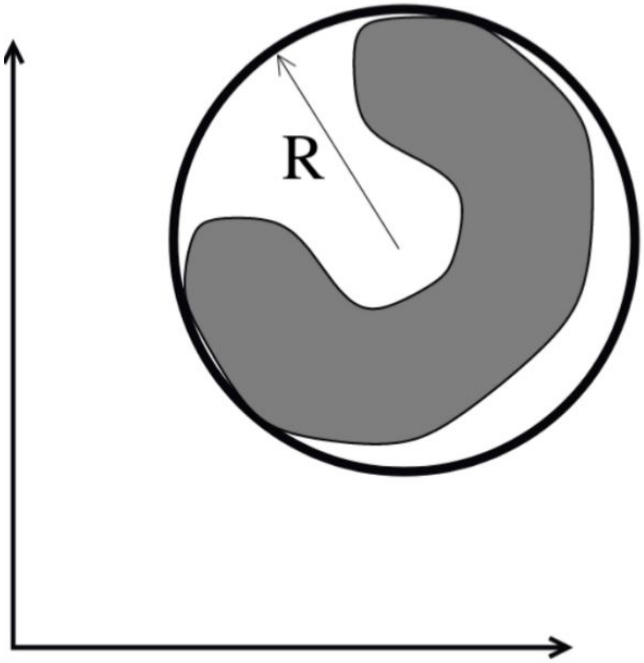
- Anomaly Detection
- DAGM : 6 classes
- Entraînement sur des exemples positifs
- Approche *Deep Learning* avec *TensorFlow*



SVDD de Tax & Duin

$$\min_{R,a,\xi} \quad R^2 + C \sum_i \xi_i$$

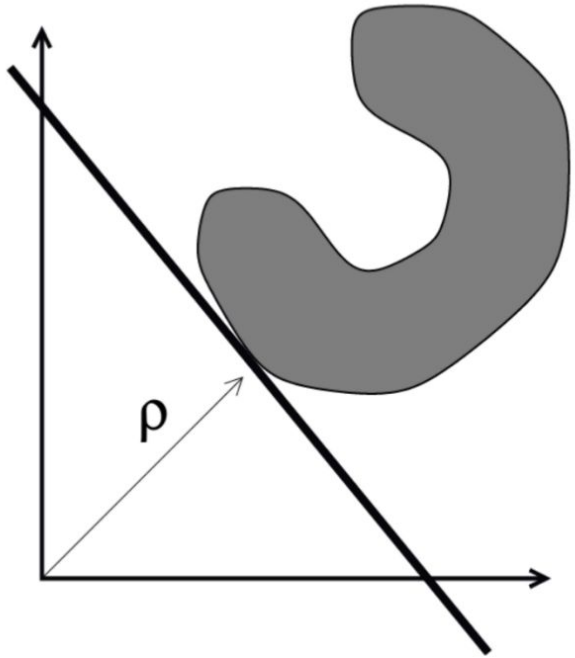
$$\forall i \quad ||\mathrm{x_i} - a||^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0.$$



OCSVM de Schölkopf

$$\min_{\mathrm{w},\rho,\xi} \quad \frac{1}{2}||\mathrm{w}||^2 - \rho + \frac{1}{\nu N}\sum_i \xi,$$

$$\forall i \quad \mathrm{w}\cdot \mathrm{x_i} \geq \rho - \xi_i, \quad \xi_i \geq 0.$$



Implémentation dans TensorFlow - SVDD

$$\xi_i = \max(||\mathrm{x_i} - a||^2 - R^2, 0),$$

$$\xi_i = -\min(R^2 - ||\mathrm{x_i} - a||^2, 0).$$

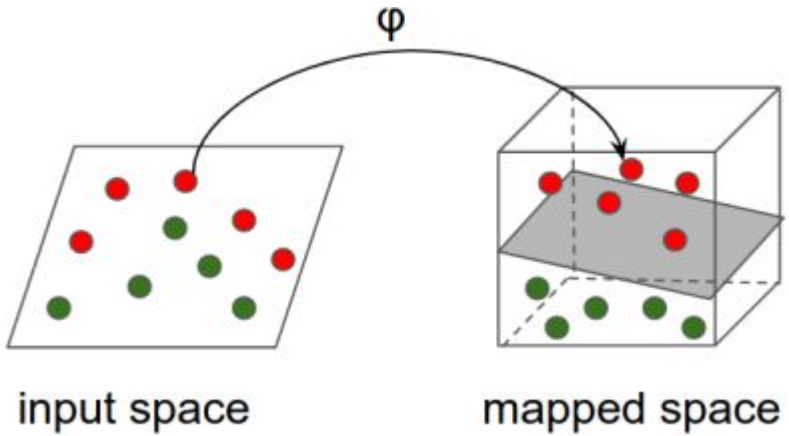
$$R^2 - \frac{C}{N}\sum_i \min(R^2 - ||\mathrm{x_i} - a||^2, 0).$$

Implémentation dans TensorFlow - OCSVM

$$\frac{1}{2}||\mathrm{w}||^2 - \rho - \frac{C}{N}\sum_i \min(\mathrm{w}\cdot \mathrm{x_i} - \rho, 0)$$

Kernels explicites

- kernel trick : map implicite
- pas de dual form : pas de kernel trick
- choix d'un map explicite
- RFFM : approximation de RBF



$$K(\mathrm{x}, \mathrm{x}') = \Phi(\mathrm{x}) \cdot \Phi(\mathrm{x}')$$

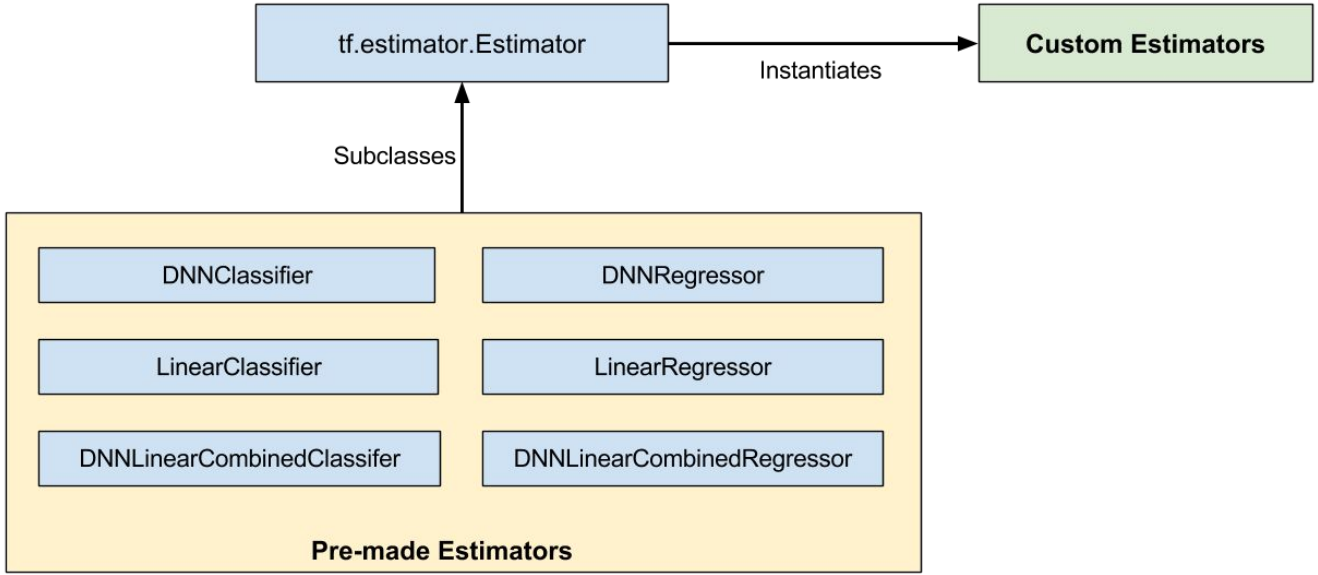
$$RFFM(\cdot) : \mathbb{R}^d \to \mathbb{R}^D, \quad RFFM(\mathbf{x}) = \cos(\boldsymbol{\Omega} \cdot \mathbf{x} + \mathbf{b})$$

$$RFFM(\mathbf{x})^T \cdot RFFM(\mathbf{y}) \approx e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$$

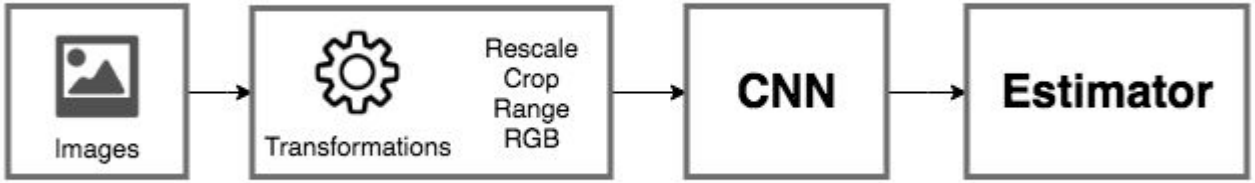
TensorFlow Estimator

Définition:
- def input_fn()
- def model_fn()

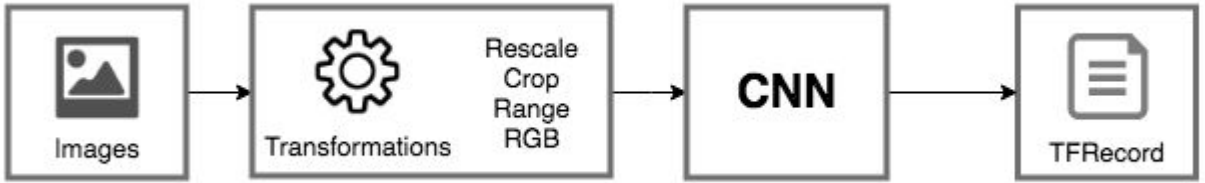
Utilisation simple:
- classifier.predict()
- classifier.train()
- classifier.evaluate()



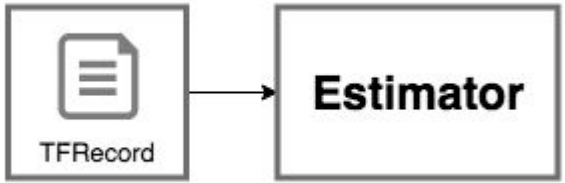
Pipeline Dataset



Mode direct



Pré-calculs et mise en cache (TFRecord)



Mode en cache

Indicateurs : définitions et matrice de confusion

— **Vrai positif** : image sans défaut prédite sans défaut
— **Faux positif** : image avec défaut prédite sans défaut
— **Vrai négatif** : image avec défaut prédite avec défaut
— **Faux négatif** : image sans défaut prédite avec défaut

Vrais négatifs	Faux positifs
Faux négatifs	Vrais positifs

Indicateurs: proportions

$$\text{Accuracy} = \frac{\text{Nombre de préditions correctes}}{\text{Nombre de prédictions totales}}$$

$$\text{Precision} = \frac{\text{Vrai positifs}}{\text{Vrai positifs} + \text{Faux positifs}}$$

$$\text{Recall} = \frac{\text{Vrai positifs}}{\text{Vrai positifs} + \text{Faux négatifs}}$$

$$\text{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Résultats avec notre implémentation

	Classe 6	Classe 5	Classe 4	Classe 3	Classe 2	Classe 1
Accuracy	65.7	56.7	78.7	55.3	62.3	56.3
Precision	65.5	55.9	87.1	54.2	67.0	55.4
Recall	66.0	63.3	67.3	68.0	48.7	65.3
F1-score	65.8	59.3	75.9	60.3	56.4	60.0
Matrice de confusion	98 | 52 51 | 99	75 | 75 55 | 95	135 | 15 49 | 101	64 | 86 48 | 102	114 | 36 77 | 73	71 | 79 52 | 98

FIGURE 6 – Résultats du SVDD en choisissant le paramètre optimal de la classe 6 ($C = 3$, kernel linéaire)

Comparaison avec Scikit-Learn

	Classe 6	Classe 5	Classe 4	Classe 3	Classe 2	Classe 1
Accuracy	65.7	56.7	78.7	55.3	62.3	56.3
Precision	65.5	55.9	87.1	54.2	67.0	55.4
Recall	66.0	63.3	67.3	68.0	48.7	65.3
F1-score	65.8	59.3	75.9	60.3	56.4	60.0
Matrice de confusion	98 52 51 99	75 75 55 95	135 15 49 101	64 86 48 102	114 36 77 73	71 79 52 98

FIGURE 6 – Résultats du SVDD en choisissant le paramètre optimal de la classe 6 ($C = 3$, kernel linéaire)

	Classe 6	Classe 5	Classe 4	Classe 3	Classe 2	Classe 1
Accuracy	65.0	53.6	55.0	53.0	46.7	51.0
Precision	69.2	53.4	56.1	53.3	46.4	51.1
Recall	54.0	58.0	46.0	48.7	43.3	48.0
F1-score	60.6	55.6	50.5	40.8	44.8	49.5
Matrice de confusion	114 36 69 81	74 76 63 87	96 54 81 69	86 64 77 73	75 75 85 65	81 69 78 72

FIGURE 7 – Résultats du `OneClassSVM` de *Scikit-Learn* en choisissant le paramètre optimal de la classe 6 ($\nu = 0.5$, kernel linéaire)

Conclusion

- Différents noyaux explicites
- TensorLayer
- Différents CNNs
- Optimisation sous contraintes
- Difficultés rencontrées