Robustesse de l'apprentissage machine face au bruit

Noé Aubin-Cadot

22 février 2020

But

But : Déterminer à quel point l'apprentissage machine est robuste en présence de bruit.

Plan

Plan :

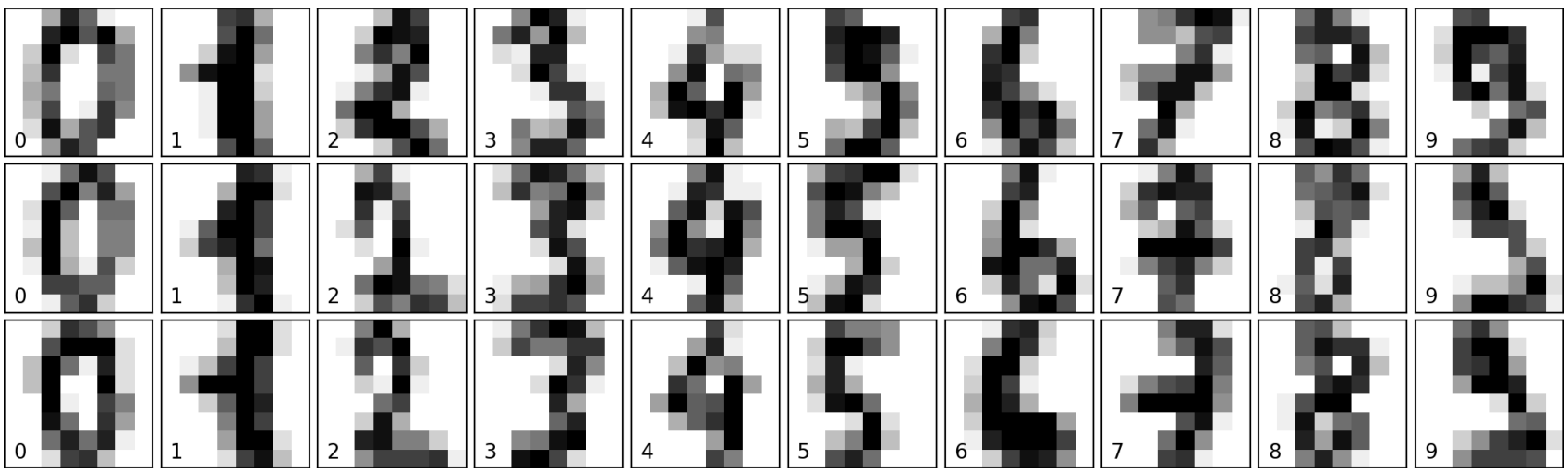
1. Trouver des données.
2. Préparer les données.
3. Visualiser les données.
4. Apprentissage machine sur les données.

Trouver des données

On considère les données `digits` de `Scikit-learn` :

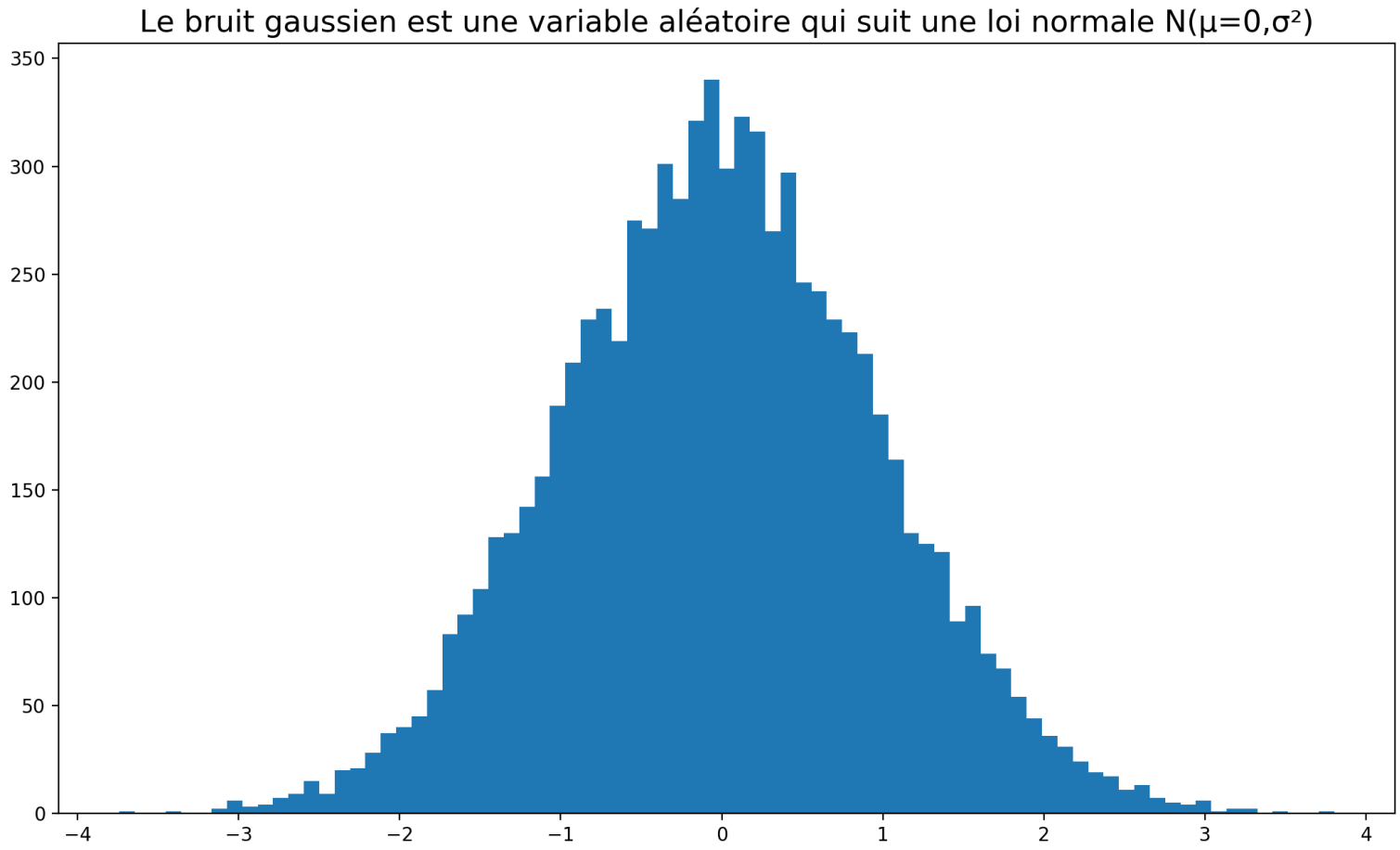
- *source* $\mathbf{X}$ = images.
- *but* $\mathbf{y}$ = chiffres $\{0, 1, 2, ..., 9\}$.

Ces données contiennent 1797 images de 8×8 pixels monochromes de 4 bits, i.e. à valeurs en $\{0, 1, 2, ..., 15\}$.



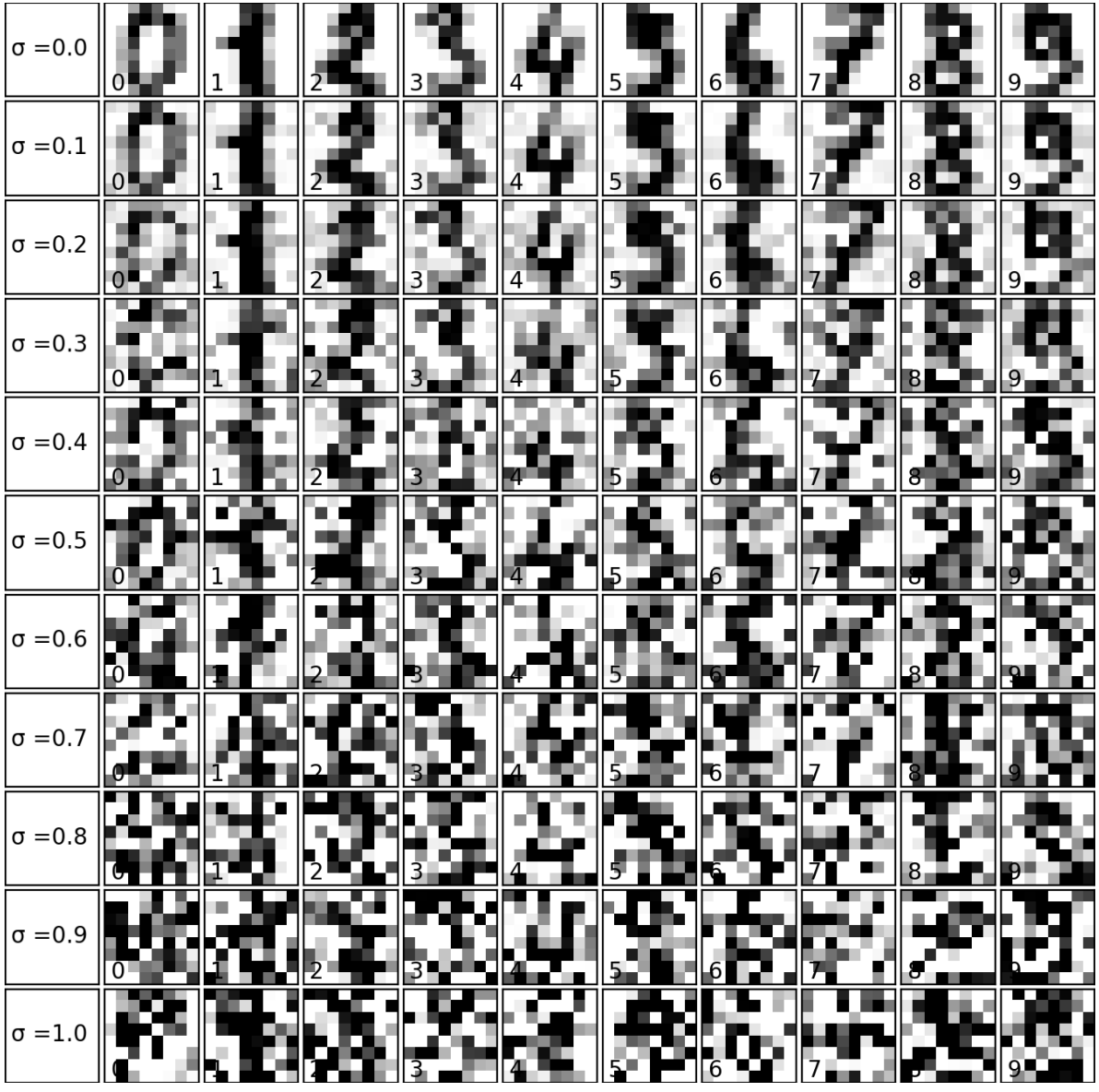
Préparer les données

On normalise les pixels pour qu'ils soient à valeurs *décimales* $[0, 1]$ au lieu qu'ils soient à valeurs *entières* $\{0, 1, 2, ..., 15\}$. Ensuite on ajoute du bruit gaussien aux pixels. Si le pixel vaut > 1, on l'égalise à 1 et s'il vaut < 0 on l'égalise à 0.



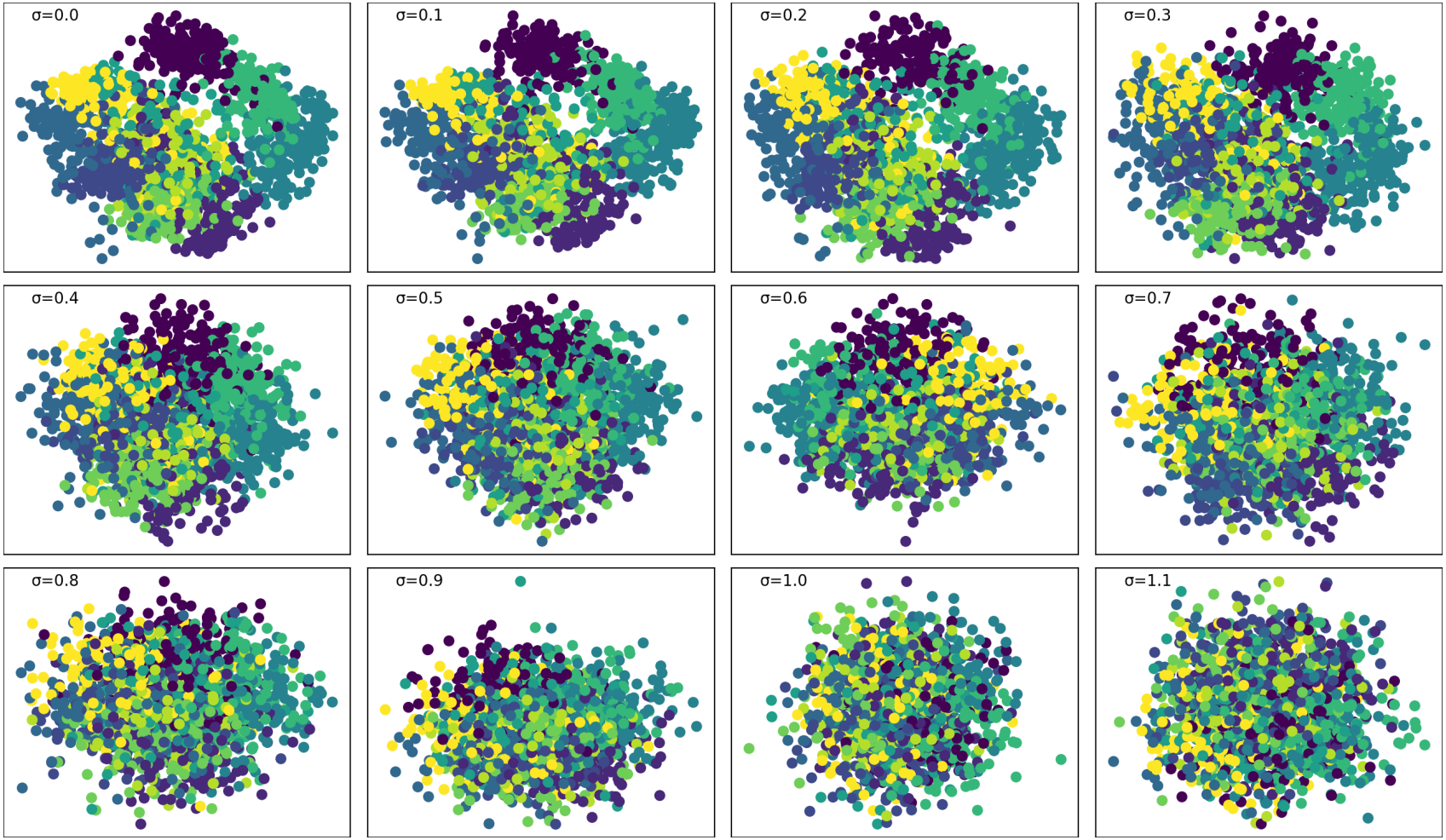
Visualiser les données

Le bruit dépend uniquement de l'écart-type σ :



Visualiser les données

Plus le bruit est grand, moins l'apprentissage machine sera performant. Voici l'analyse en composantes principales (PCA) des données bruitées pour $\sigma \in \{0.0, 0.1, 0.2, ..., 1.1\}$:



Apprentissage machine sur les données

On scinde les données $(\mathbf{X}, \mathbf{y})$ en deux sous-ensembles :

- 75% : *entraînement* $(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$, 1347 images.
- 25% : *test* $(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$, 450 images.

On entraîne un classificateur sur les données d'entraînement et on évalue ses résultats sur les données de test.

On peut essayer divers classificateurs `scikit-learn` e.g. KNN, BNG, BNB, SVM, lbfgs, liblinear, RFC, Perceptron, SGDC, DTC, etc.

Apprentissage machine sur les données

Scores d'apprentissage sur les images non bruitées (i.e. $\sigma = 0$) :

Nom	Train	Test
KNN	100.0%	98.2%
BNG	87.6%	85.6%
BNB	86.3%	87.3%
SVM	99.2%	98.2%
lbf	98.4%	96.2%
lib	97.8%	96.7%
RFC	100.0%	93.6%
Per	97.2%	95.1%
SGD	98.5%	96.0%
DTC	100.0%	85.1%

Apprentissage machine sur les données

Matrice de confusion (i, j) = (réel, prédit) pour classificateur SVM sur les données non bruitées :

	0	1	2	3	4	5	6	7	8	9
0	39	0	0	0	0	0	0	0	0	0
1	0	37	0	0	0	0	0	0	0	0
2	0	0	43	0	0	0	0	0	0	0
3	0	0	0	45	0	1	0	0	0	0
4	0	0	0	0	49	0	0	0	1	0
5	0	0	0	0	0	48	1	0	0	0
6	0	0	0	0	0	1	51	0	0	0
7	0	0	0	0	0	0	0	38	0	0
8	0	2	0	1	0	0	0	0	42	0
9	0	0	0	0	0	1	0	0	0	50

8 mauvaises classifications sur 450 prédictions.

Apprentissage machine sur les données

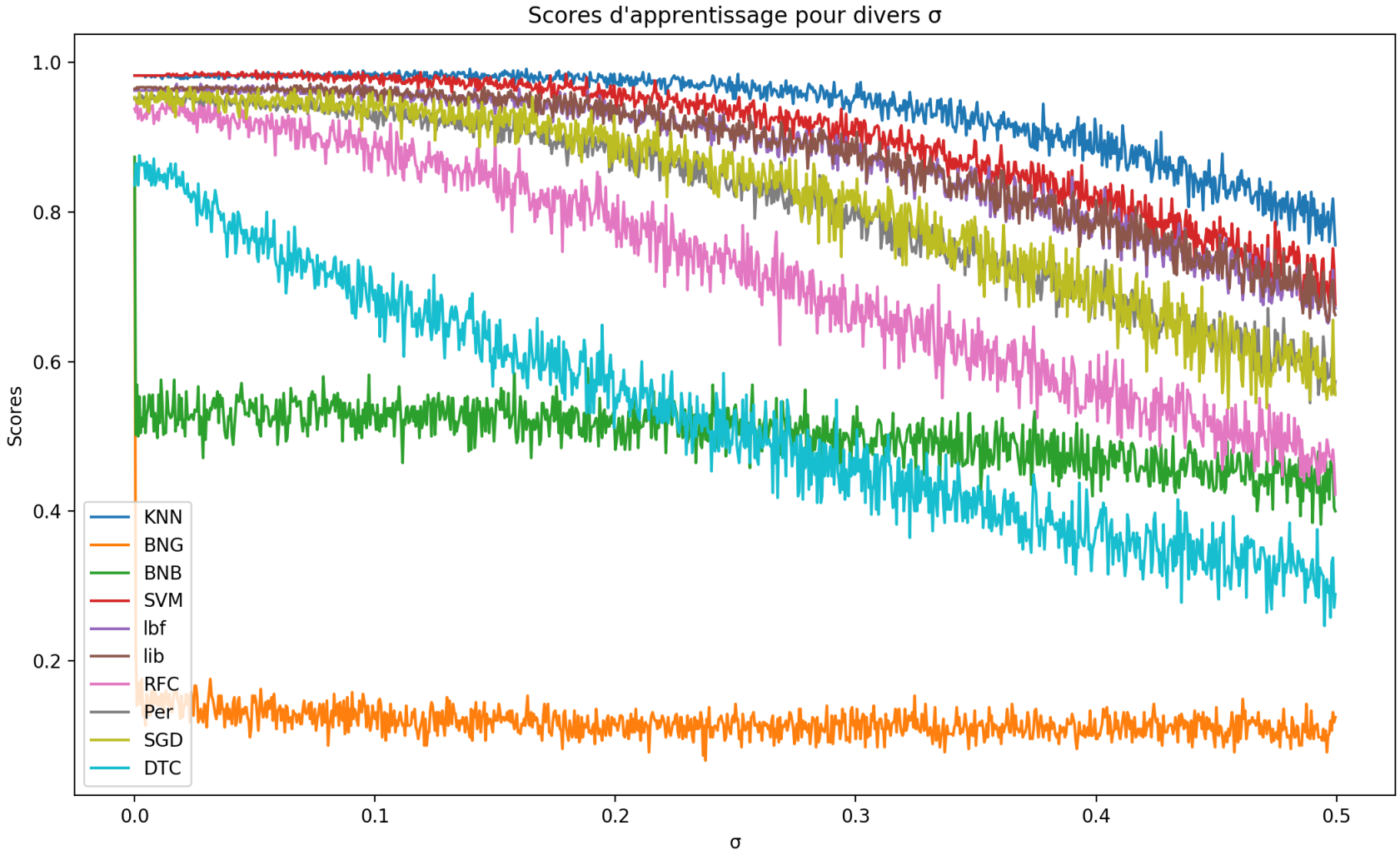
On ajoute maintenant du bruit aux images.

Deux manières intéressantes d'étudier la robustesse de l'apprentissage machine face au bruit :

1. Bruiter $\mathbf{X}_{\text{test}}$ mais non $\mathbf{X}_{\text{train}}$.
2. Bruiter $\mathbf{X}_{\text{test}}$ et $\mathbf{X}_{\text{train}}$.

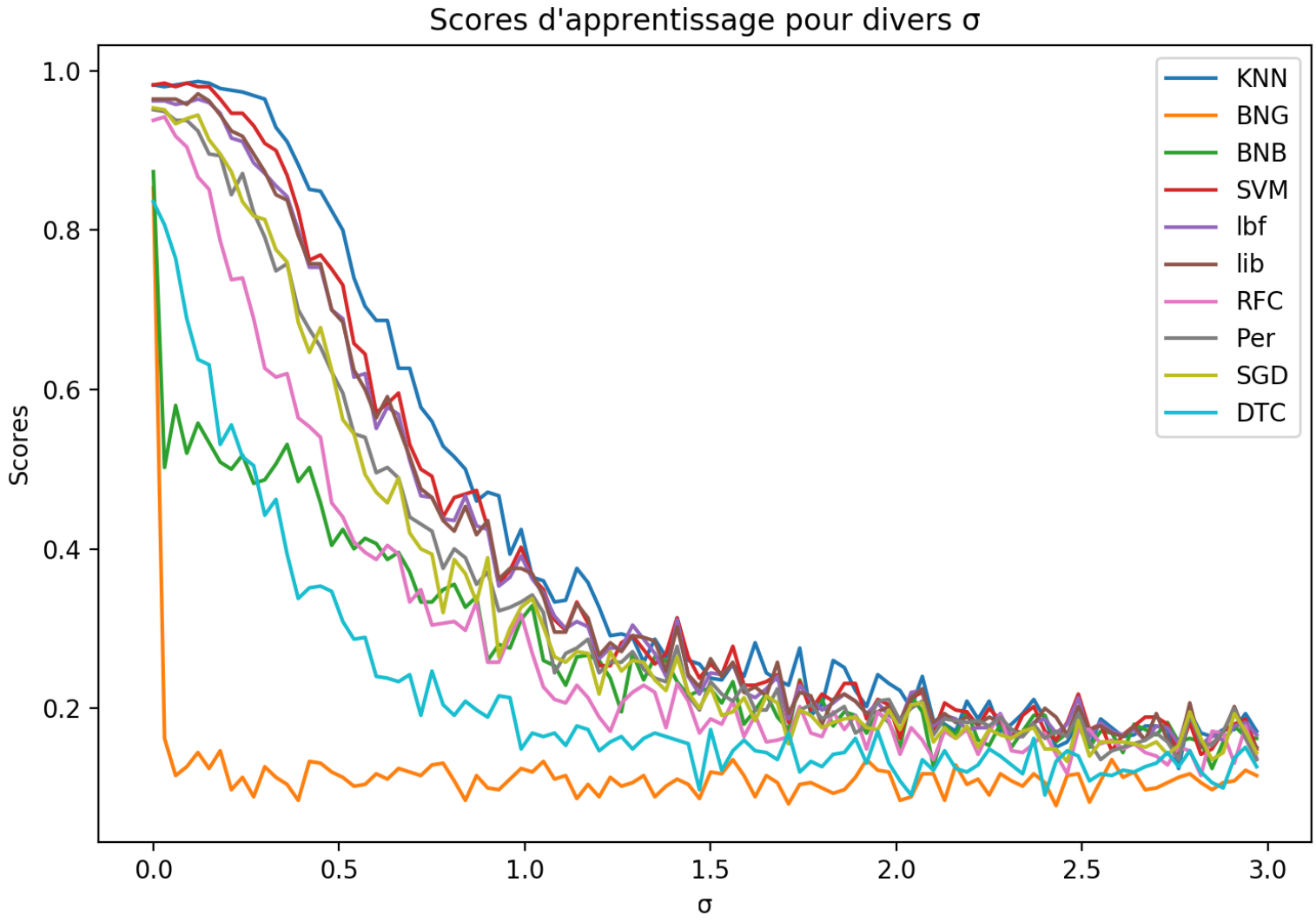
1. $\mathbf{X}_{\text{test}}$ bruité mais $\mathbf{X}_{\text{train}}$ non bruité

Pour $\mathbf{X}_{\text{test}}$ bruité mais $\mathbf{X}_{\text{train}}$ non bruité l'algorithme KNN ($k = 1$) est le plus robuste face au bruit :



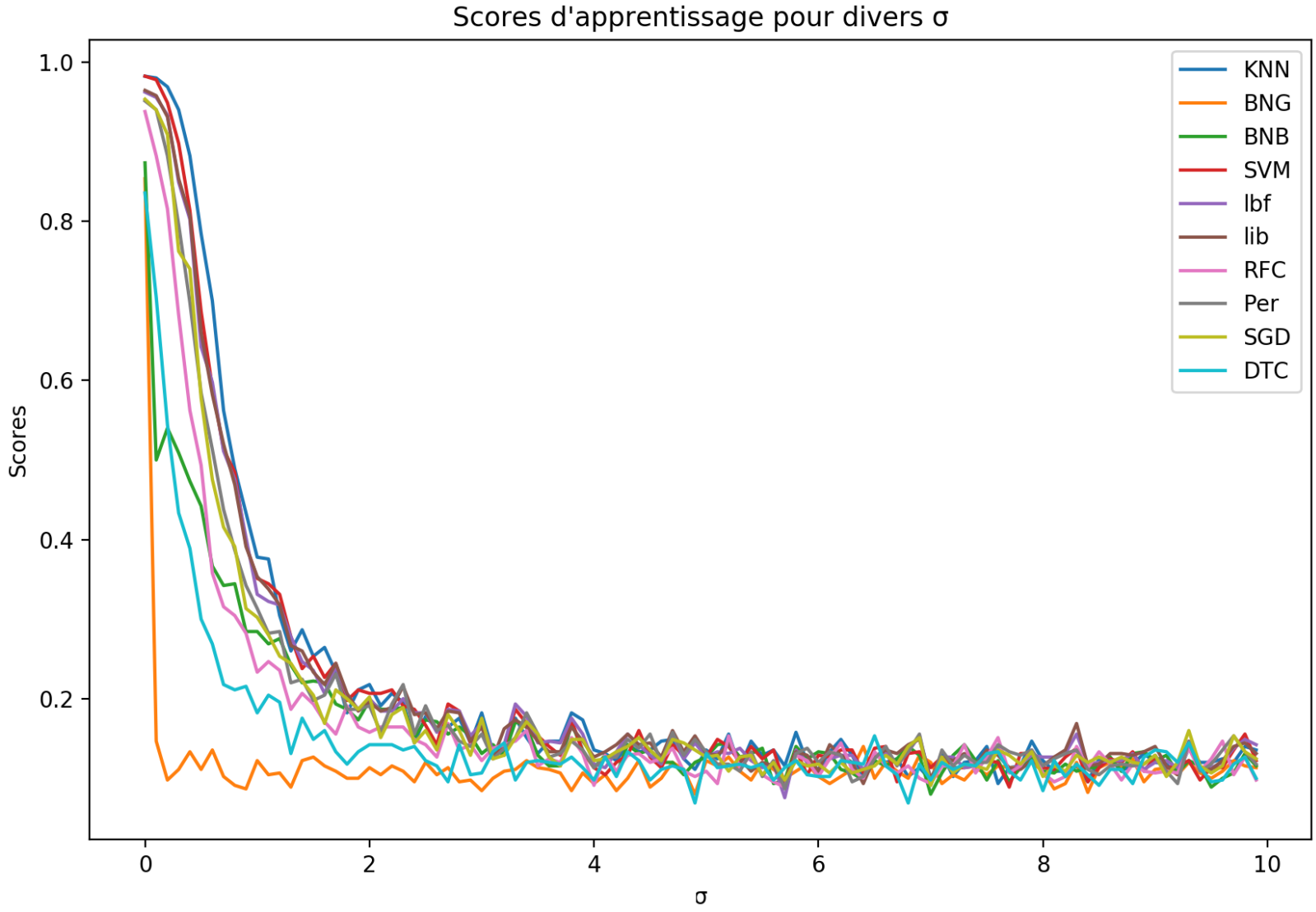
1. X_{test} bruité mais X_{train} non bruité

On peut regarder sur un plus grand domaine pour σ :



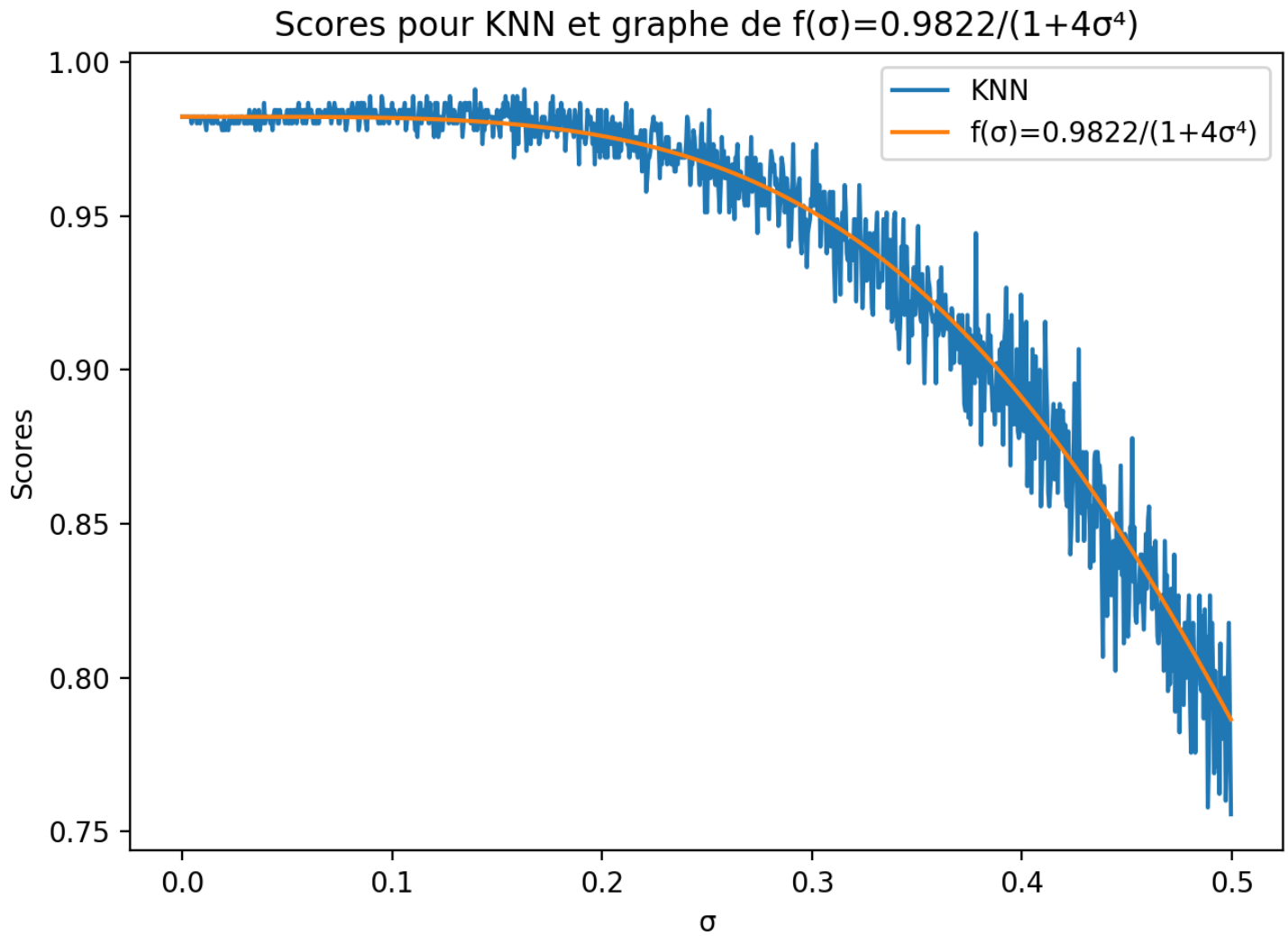
1. X_{test} bruité mais X_{train} non bruité

Quand $\sigma \to \infty$, les scores tendent à 1 chance sur 10 :



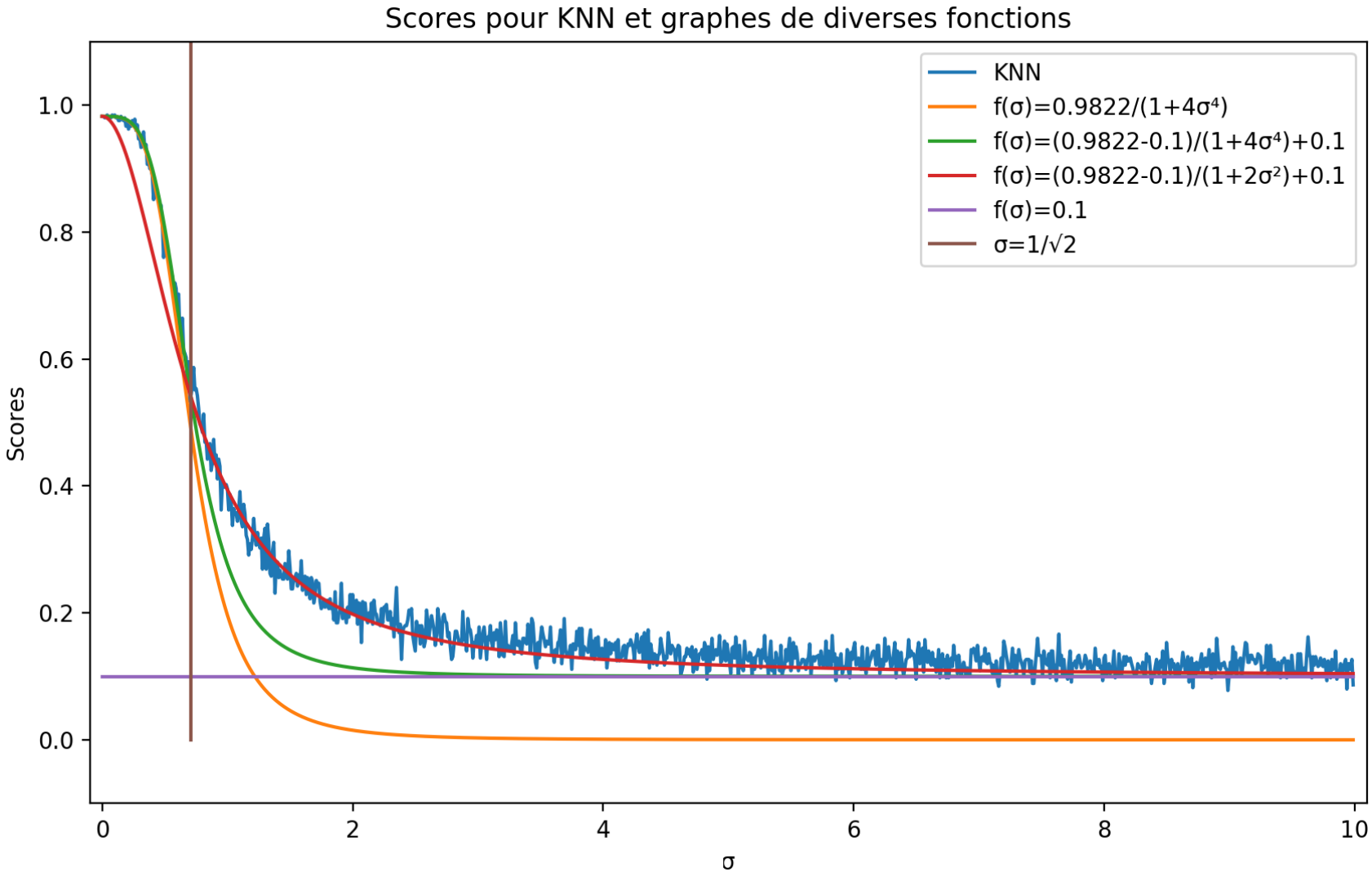
1. X_{test} bruité mais X_{train} non bruité

Pour $\sigma = 0$, le score de KNN était de 98.22%. Pour σ petit, les scores de KNN suivent $f(\sigma) = 0.9822/(1 + 4\sigma^4)$:



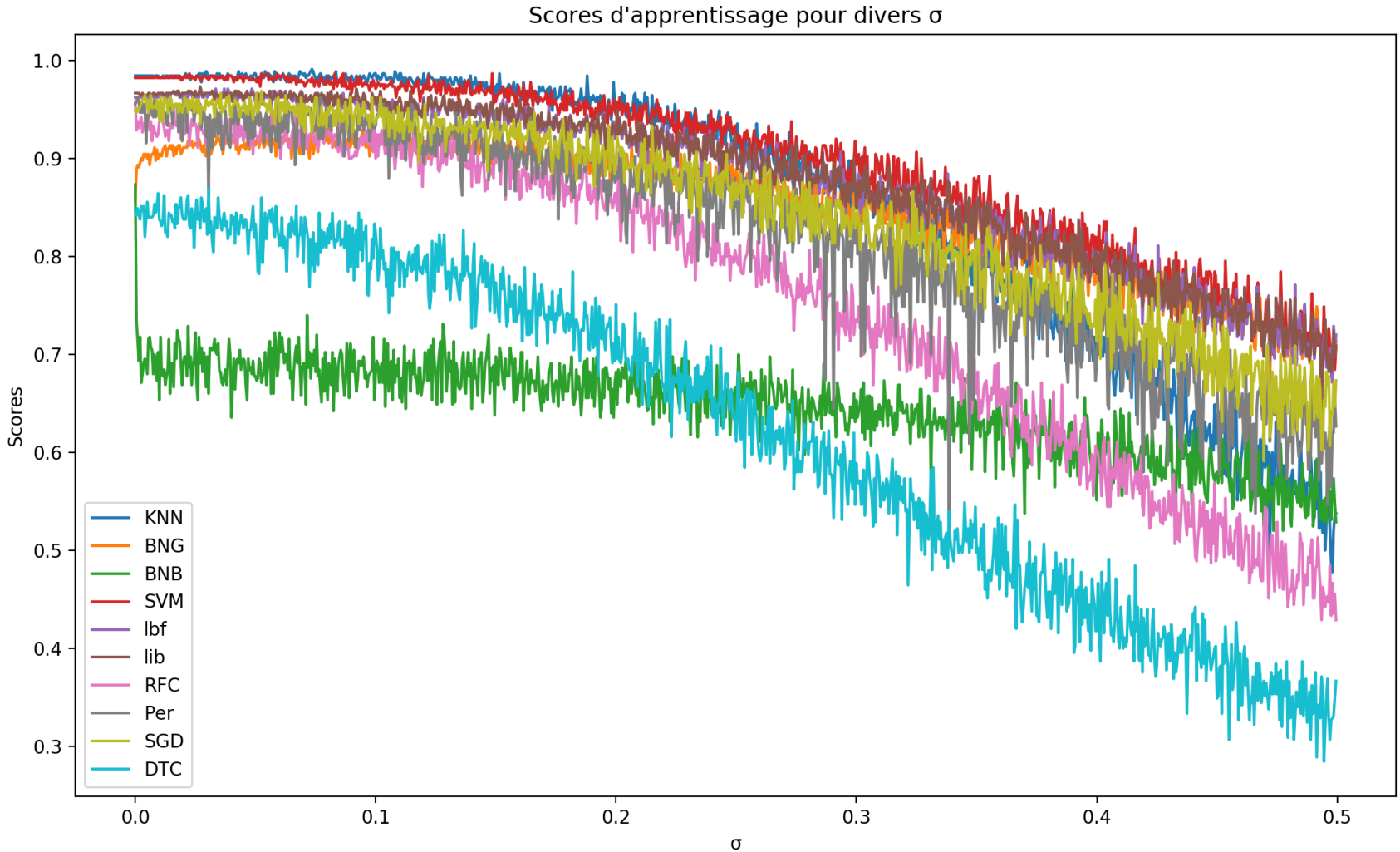
1. X_{test} bruité mais X_{train} non bruité

Pour $\sigma \leq 1/\sqrt{2}$ le score suit $f(\sigma) = 0.9822/(1+4\sigma^4)$, pour $\sigma \geq 1/\sqrt{2}$ le score suit $f(\sigma) = (0.9822-0.1)/(1+2\sigma^2)+0.1$:



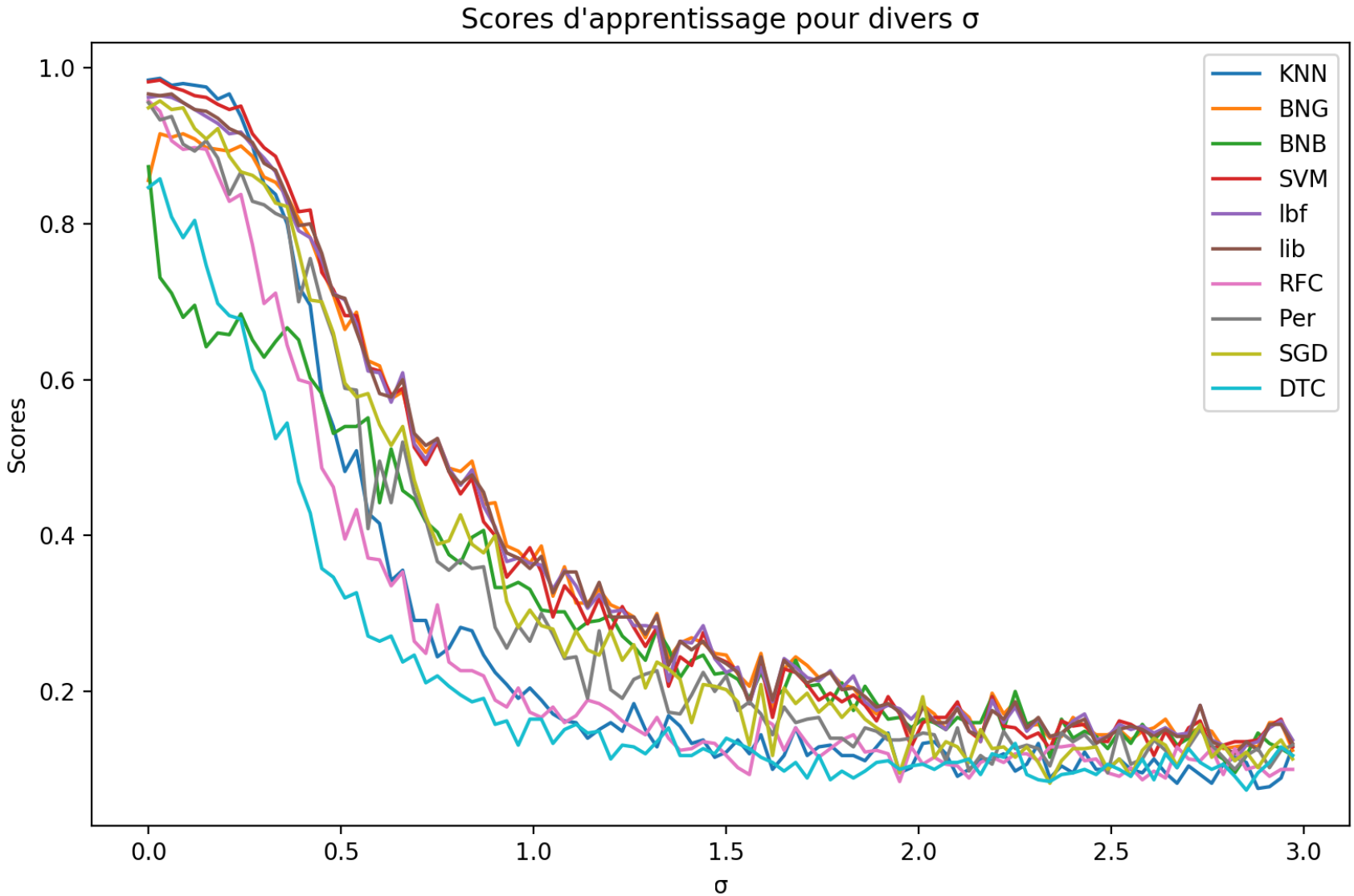
2. X_{test} et X_{train} bruités

Maintenant on bruite X_{test} et X_{train}. KNN ne domine plus les scores. Ici c'est SVM qui domine :



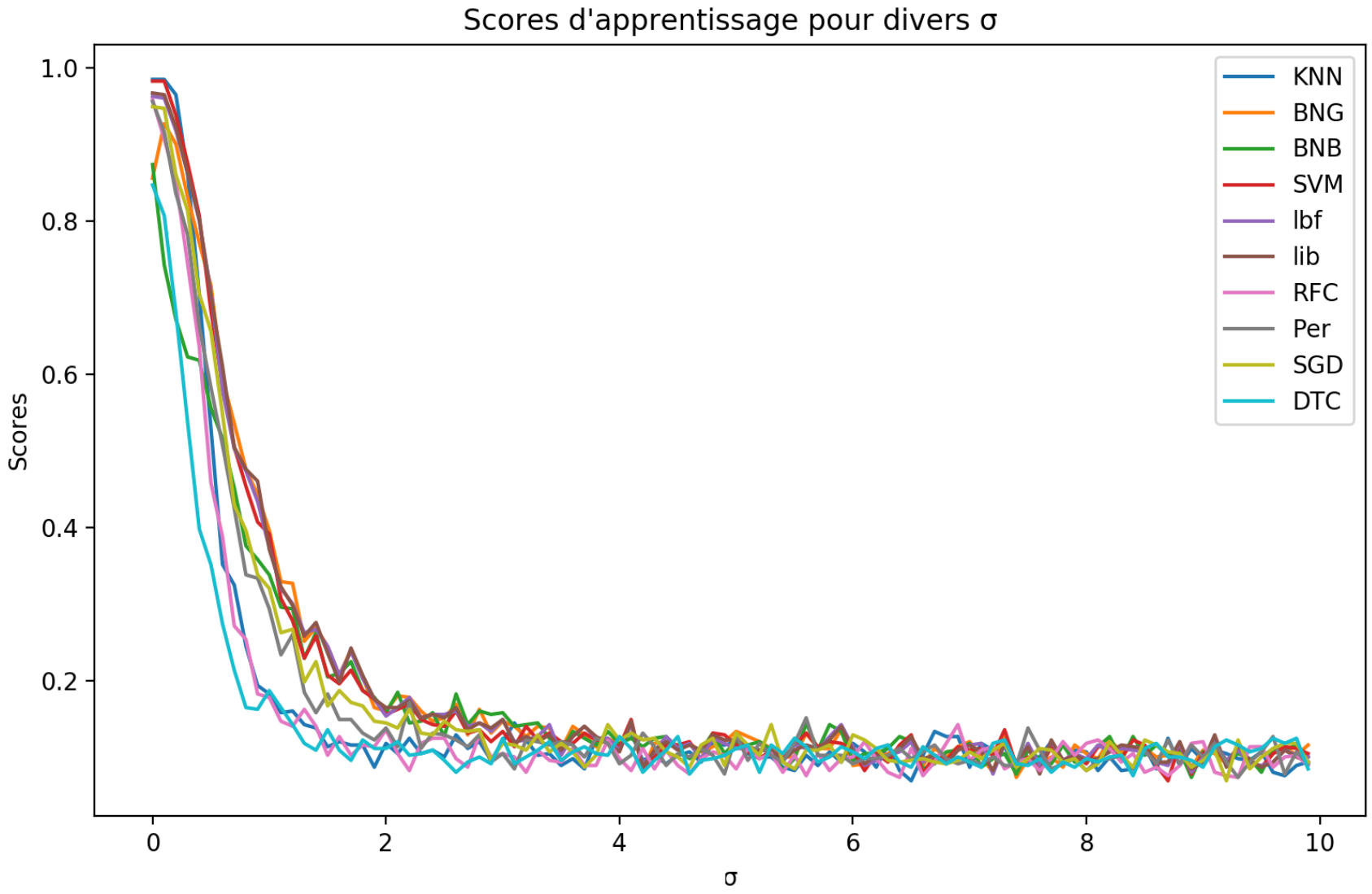
2. X_{test} et X_{train} bruités

On peut regarder sur un plus grand domaine :



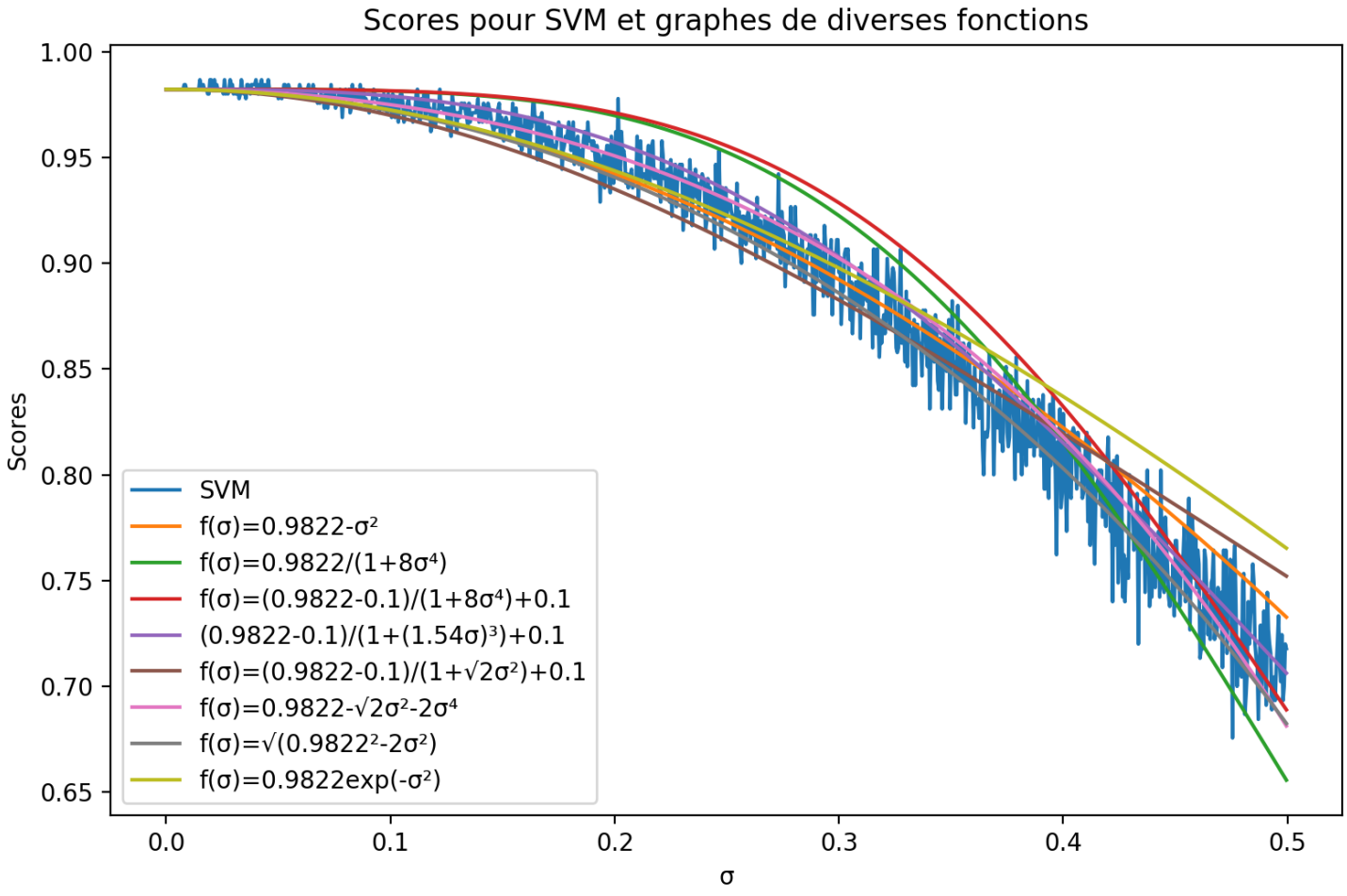
2. X_{test} et X_{train} bruités

Et sur un domaine encore plus grand :



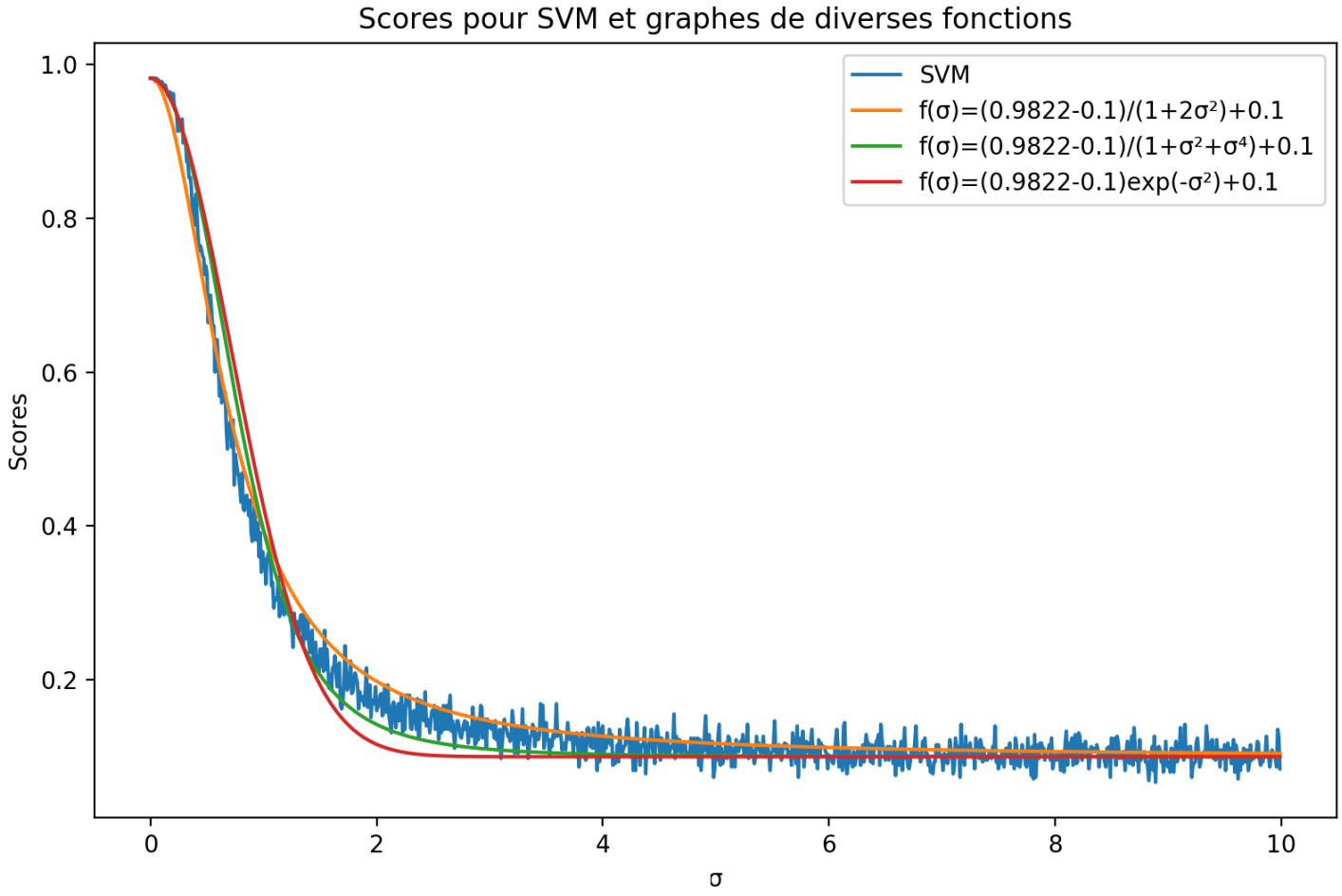
2. X_{test} et X_{train} bruités

Pour σ petit il est difficile de *fitter* une fonction sur les scores de SVM :



2. X_{test} et X_{train} bruités

Pour σ grand il est tout aussi difficile de *fitter* une fonction sur les scores de SVM :



Merci de votre attention ☺