Détection de zones par montée de gradient d'un rectangle dans le flou gaussien

Noé Aubin-Cadot

22 mars 2020

But

But : Détecter des régions statistiquement peu probables dans un nuage de points à valeurs binaires.

Idée : À l'aide d'un noyau gaussien, construire une fonction plutôt lisse sur laquelle un rectangle glisse en suivant une montée de gradient vers une région où il se passe quelque chose.

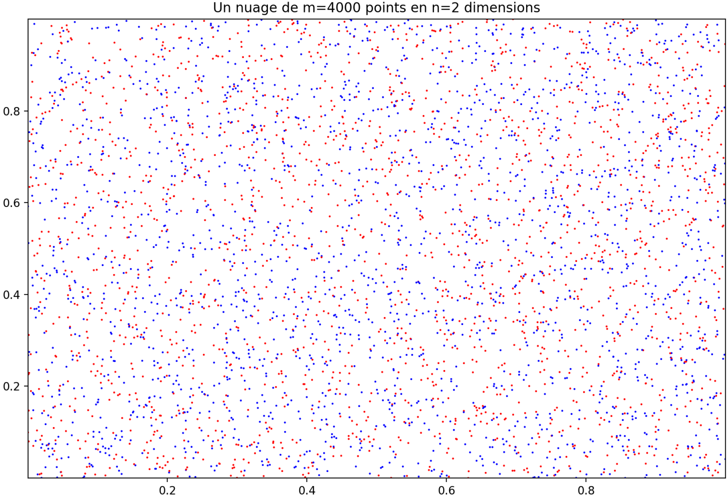
Plan

Plan :

1. Visualiser le problème.
2. Algorithme de détections.
3. Utilisation de l'algorithme.
4. Réflexions à propos de l'algorithme.

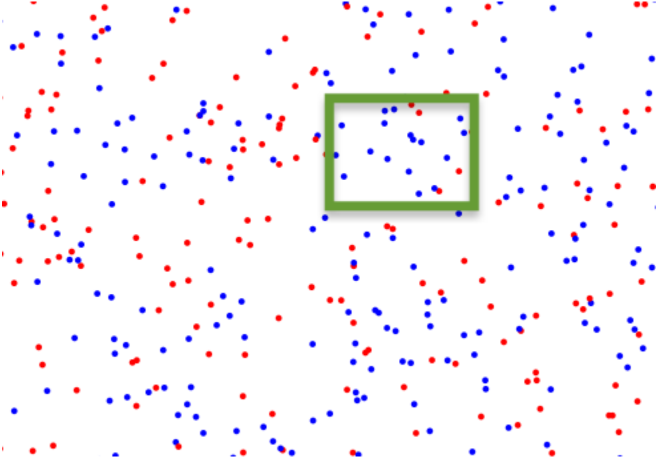
Visualiser le problème

On considère un nuage de m points dans un espace de dimension n. Chaque point est à valeurs binaires $\{0, 1\}$.



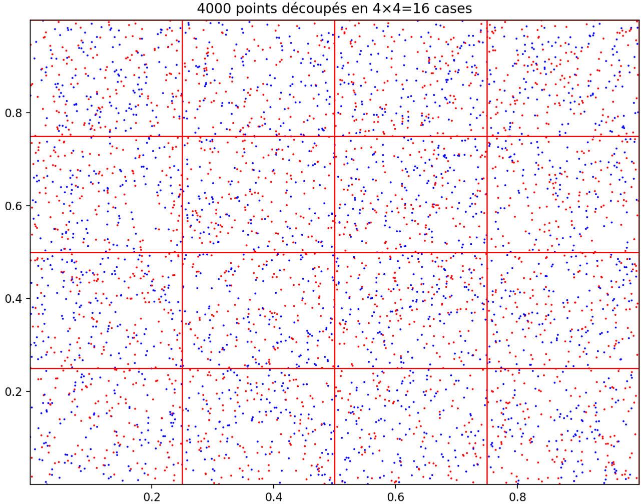
Visualiser le problème

On cherche des régions où la proportion locale de `0` et de `1` est éloignée de la proportion globale de `0` et de `1`.



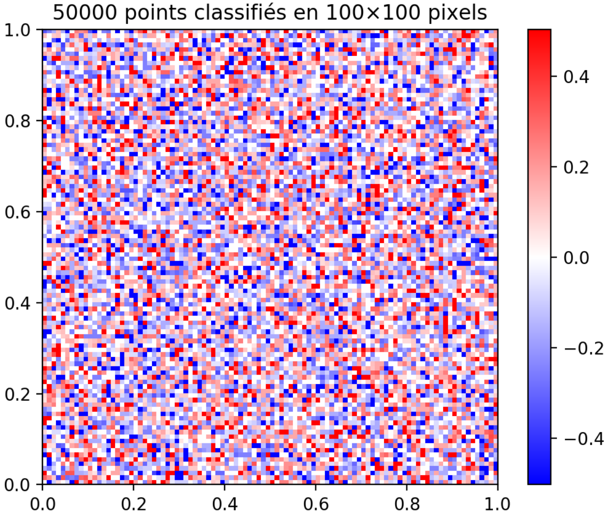
Algorithme de détections

D'abord, on découpe le nuage de points selon une grille de résolution $k_1 \times k_2 \times \ldots \times k_n$.



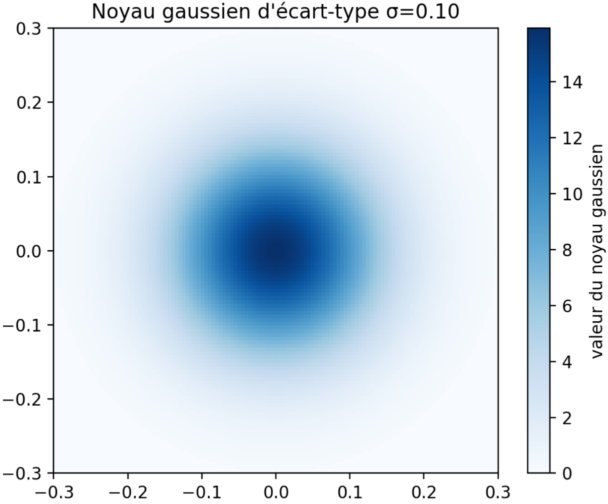
Algorithme de détections

On voit chaque case comme un pixel dont la couleur est la proportion locale de `0` et de `1` moins la proportion globale.



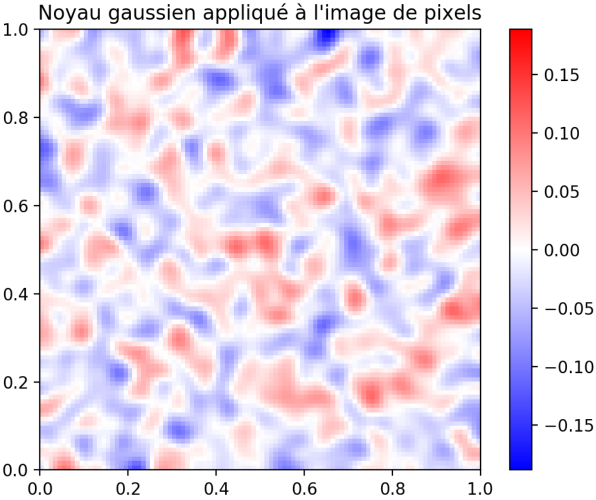
Algorithme de détections

On fait alors une convolution avec un noyau gaussien :



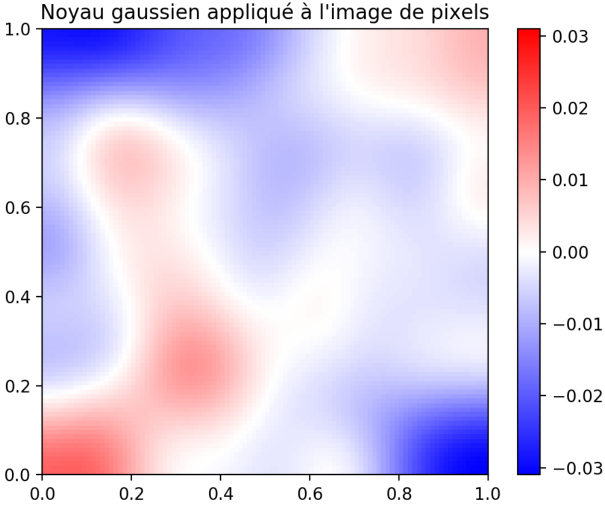
Algorithme de détections

La nouvelle image obtenue est plus lisse :



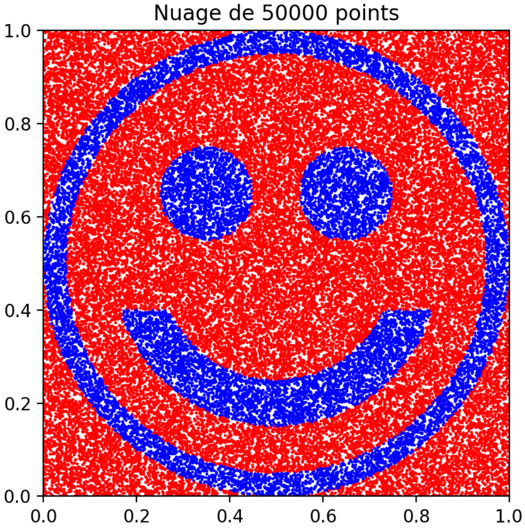
Algorithme de détections

En modifiant l'écart-type σ du noyau gaussien on peu effacer ou garder plus ou moins de détails locaux :



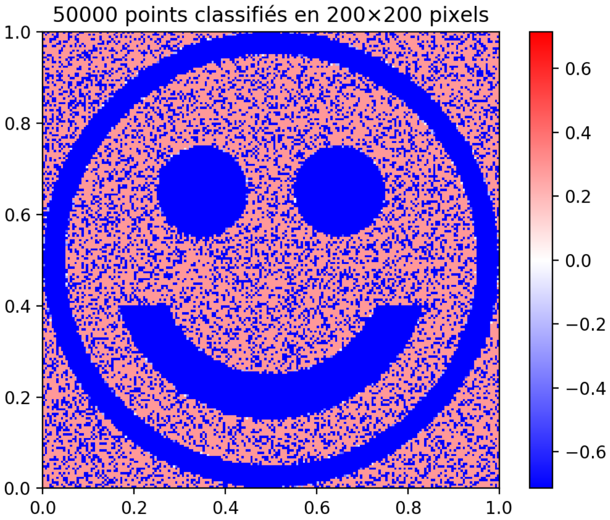
Algorithme de détections

Considérons cet autre nuage de points :



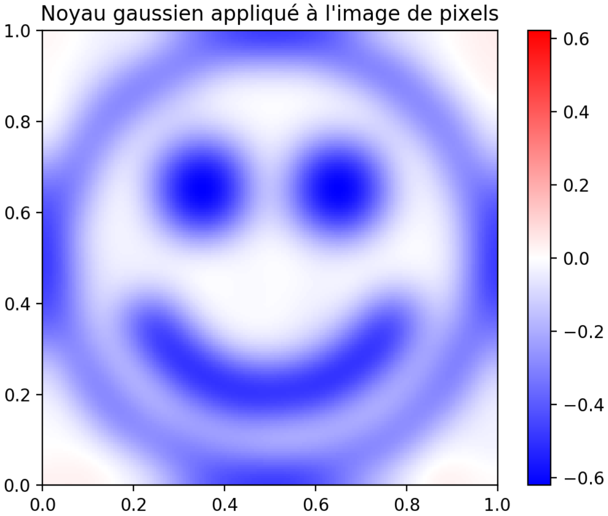
Algorithme de détections

On classifie les points dans des pixels :



Algorithme de détections

On y applique le noyau gaussien :

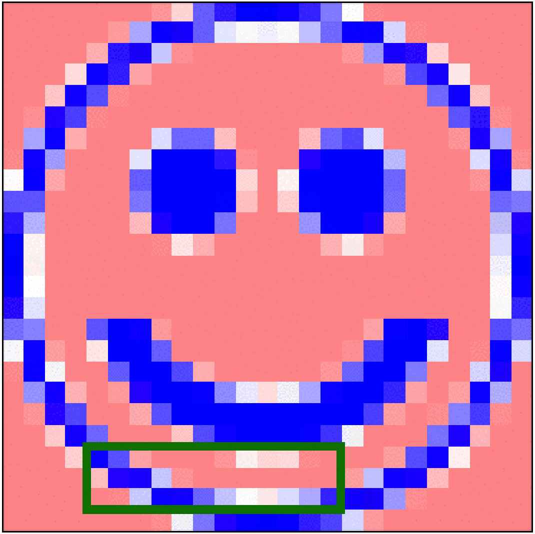


Algorithme de détections

Maintenant qu'on a une image lisse, un rectangle peut suivre le gradient pour trouver des régions où la proportion locale est éloignée de la proportion globale.

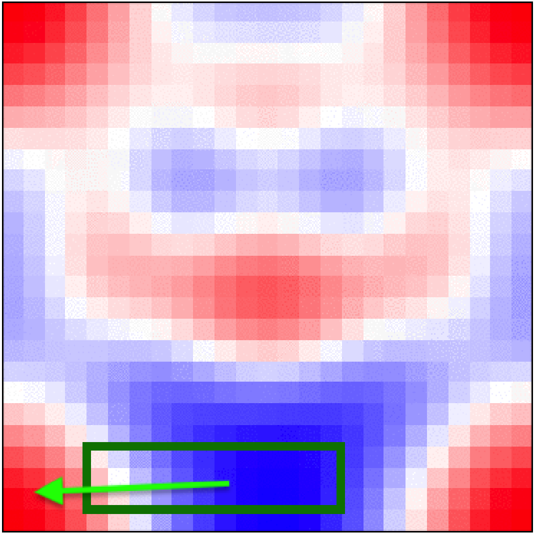
Utilisation de l'algorithme

On commence avec un rectangle en position aléatoire :



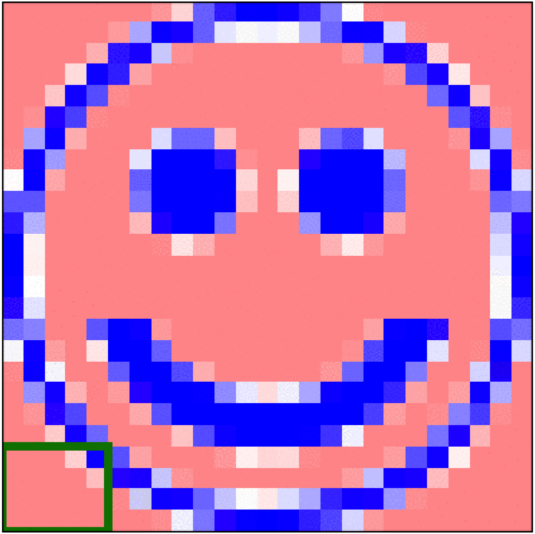
Utilisation de l’algorithme

On laisse aller le rectangle dans le gradient du flou gaussien :



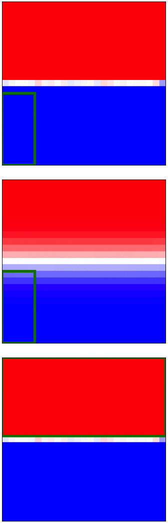
Utilisation de l'algorithme

On constate où le rectangle a convergé :



Utilisation de l'algorithme

Le flou gaussien permet de détecter des régions éloignées :



Réflexions à propos de l'algorithme

En dimension $n = 2$ l'algorithme est très rapide, même pour un très grand nombre m de points.

Il est aussi possible de l'étendre aux données incomplètes. Un "point" dont on ne sait certaines coordonnées est un sous-espace. Un sous-espace est envoyé, via le noyau gaussien, à une gaussienne qui est constante dans les directions inconnues.

Réflexions à propos de l'algorithme

Toutefois, en dimension $n \gg 2$, l'application numérique du noyau gaussien devient très lourd en calculs et en mémoire.

Par exemple, en dimension $n = 600$, pour une résolution de 10 dans chaque dimension, on doit s'occuper de 10^{600} cases, ce qui est bien plus grand que $\approx 10^{80}$, le nombre d'atomes dans l'univers.

Merci de votre attention ☺