

Application de l'optimisation à la classification binaire d'image: cas des modèles SVM

Sous la supervision de Dr Diop, Juin 2024

Table des matière



Intérêt du sujet

Une présentation général du sujet



Problème de classification et présentation de SVM

présentation de techniques d'extraction de caractéristiques d'une image puis présentation de SVM.



Formulation mathématique du problème et cas pratique

Formulation mathématique et présentations d'un cas pratique sous python.

1

Intérêt du sujet

Présentation générale

La reconnaissance d'image

Dichotomie homme-machine

En tant qu'être humain, nous sommes capables, en un coup d'œil, de distinguer un chien d'un chat. Vous êtes-vous déjà demandé comment cela se passe ?

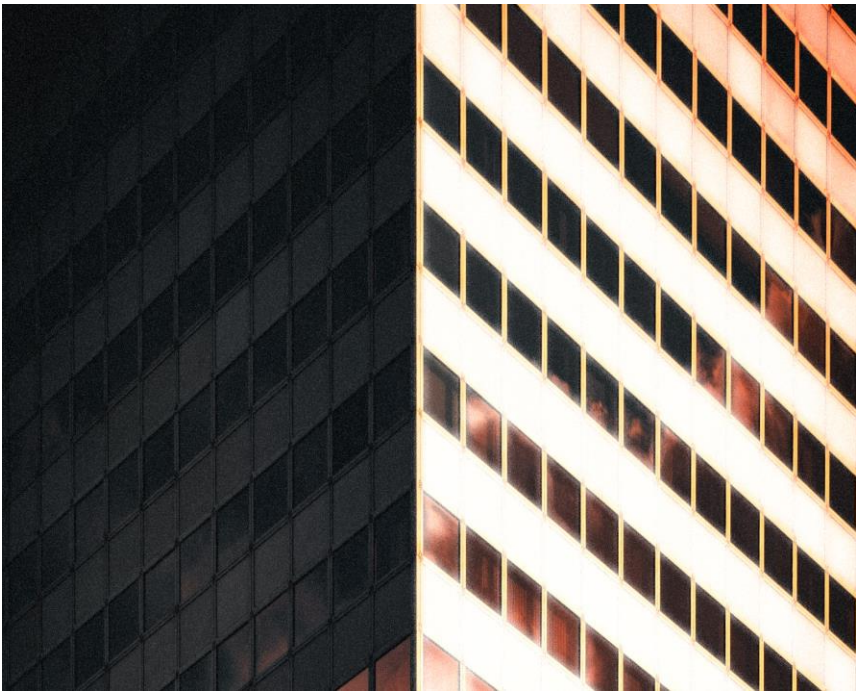
Notre cerveau traite des millions d'informations visuelles en une fraction de seconde, nous permettant d'identifier des objets et des êtres vivants avec une précision étonnante.

Cette capacité naturelle repose sur un réseau complexe de neurones qui interprète les formes, les couleurs, les textures, et d'autres indices visuels pour reconnaître les objets dans notre environnement.



La reconnaissance d'image

Dichotomie homme-machine



Dans le domaine de l'informatique, reproduire cette capacité de reconnaissance visuelle est un défi majeur qui a conduit au développement de diverses techniques et algorithmes de vision par ordinateur.

Mais comment un ordinateur conçoit-il une image ? Contrairement à notre perception intuitive, un ordinateur traite les images sous forme de données numériques.

La reconnaissance d'image

Dichotomie homme-machine

Une image numérique est représentée par **une matrice de pixels**. Chaque pixel est un petit point qui porte plusieurs informations clés : sa position, ses composantes de couleur, et parfois sa transparence.



Une image

Position

Chaque pixel a des coordonnées (x, y) qui déterminent son emplacement dans la grille de l'image.

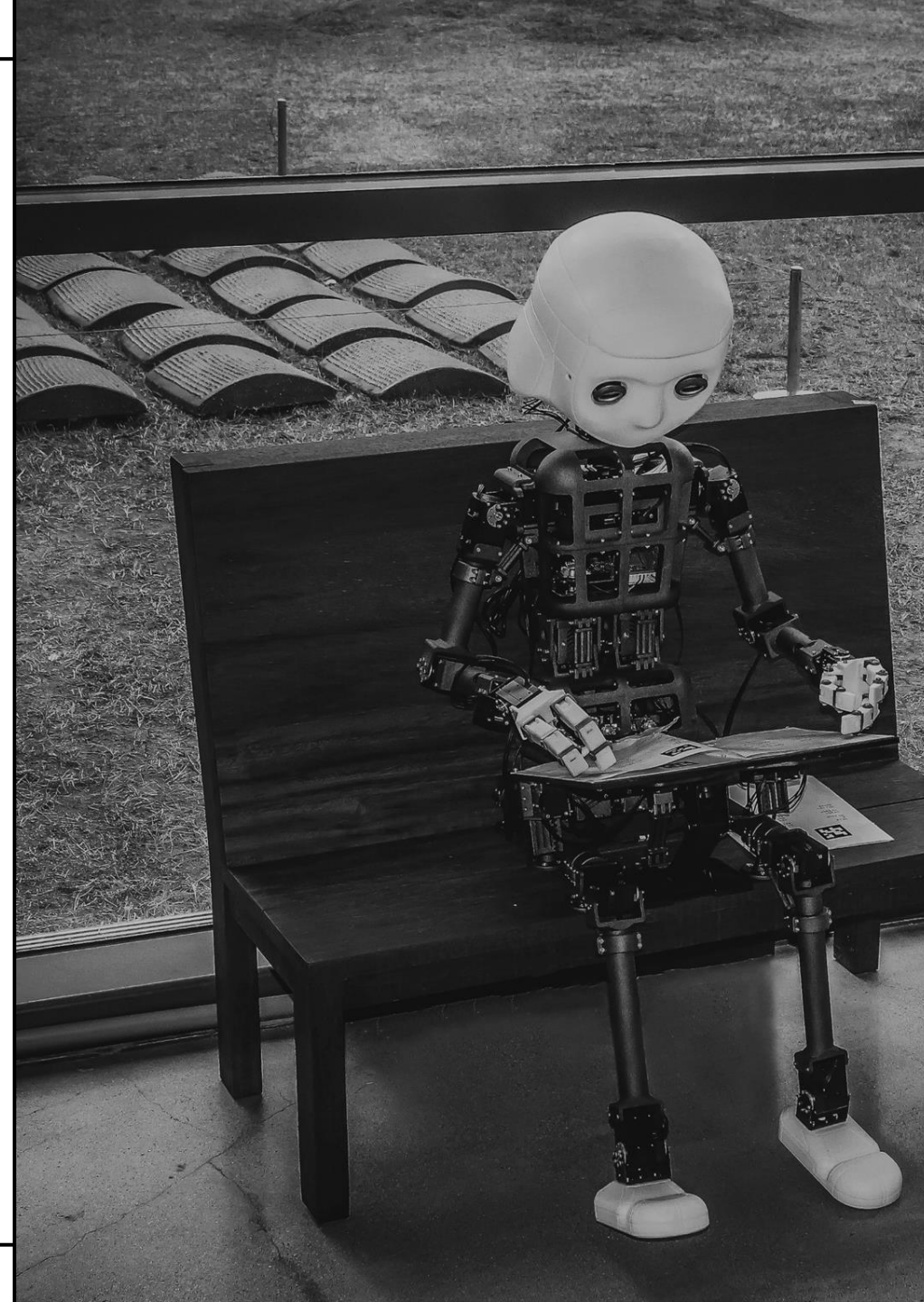
couleur

chaque pixel est représenté par trois valeurs correspondant aux composantes rouge, vert, et bleu (RVB). Par exemple, une valeur de (255, 0, 0) représente un pixel rouge pur.

Transparence

Certains formats d'image incluent une valeur de transparence. Cela permet de rendre les pixels partiellement ou totalement transparents.

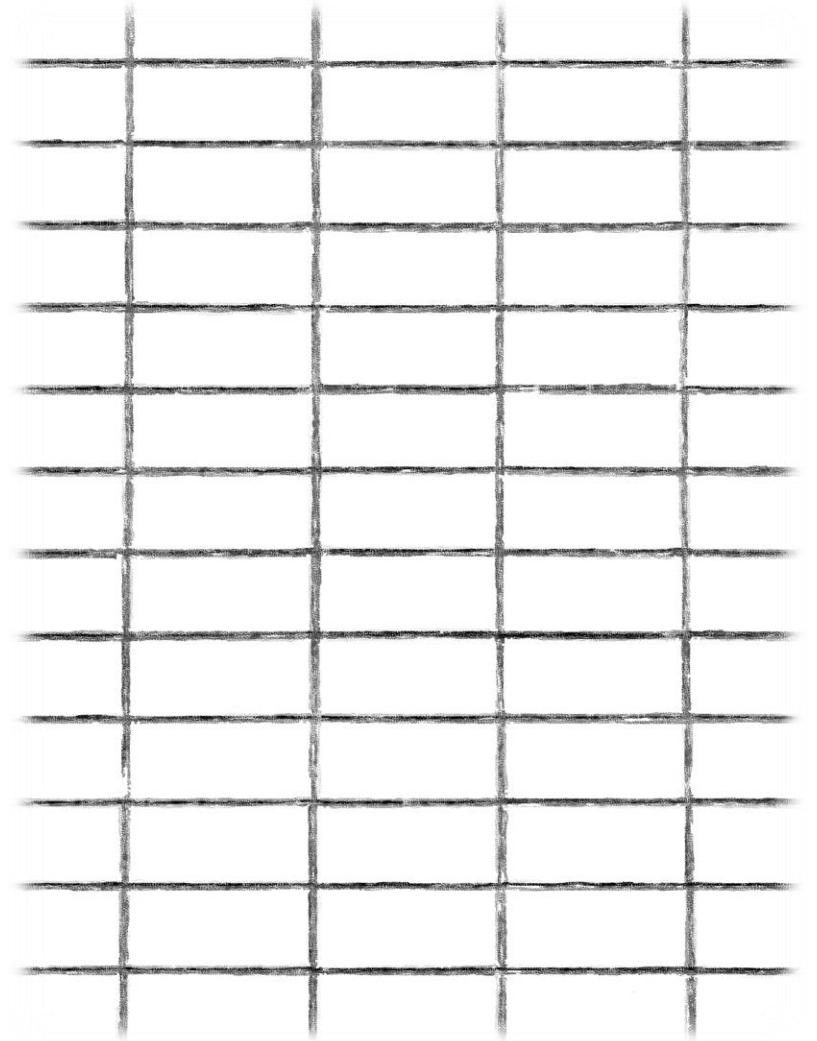
La reconnaissance d'images,
c'est le processus par lequel un
ordinateur peut regarder une
image et dire ce qu'elle
contient. Par exemple,
reconnaître si une photo
contient un chat ou non.



La reconnaissance d'image

Dichotomie homme-machine

Pour un ordinateur, analyser et comprendre une image revient à traiter cette **grande matrice de valeurs numériques**. Les algorithmes de vision par ordinateur utilisent ces valeurs pour extraire des caractéristiques pertinentes (comme les contours, les textures, et les formes) qui peuvent être utilisées pour identifier et classer les objets dans l'image.



Diagnostic Médical avec des Images Médicales

Les SVM aident à classifier les
images médicales pour détecter
des maladies.

Exemple : détection de cellules
cancéreuses dans des images de biopsies.





Surveillance Agricole et Détection des Maladies des Cultures

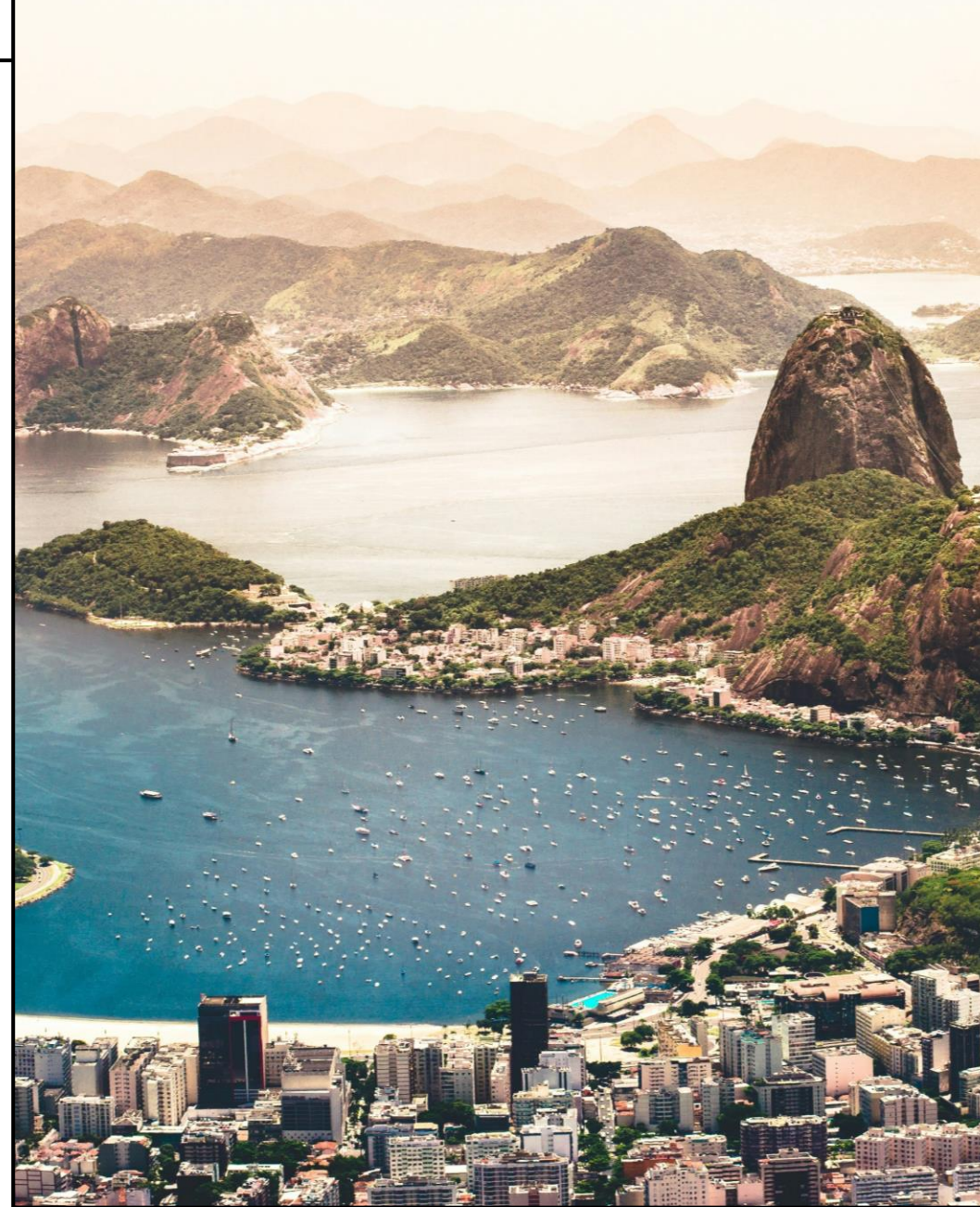
Les SVM peuvent classer les images de cultures pour identifier les maladies des plantes ou les infestations d'insectes.

Exemple : détection de maladies du maïs ou du manioc à partir d'images prises par des drones ou des smartphones..

Amélioration des Réponses aux Urgences Humanitaires

Analyse d'images satellites pour évaluer les dommages causés par des catastrophes naturelles comme les inondations ou les sécheresses.

Exemple : identification des zones les plus touchées pour une distribution efficace des aides humanitaires.



2

Problème de **classification et présentation de SVM**

Présentation générale de VGG16 et SVM

La reconnaissance d'image

Extraction de caractéristiques d'une image

Maintenant que nous savons ce qu'est une image pour un ordinateur, penchons-nous sur la question de l'extraction des caractéristiques de l'image.

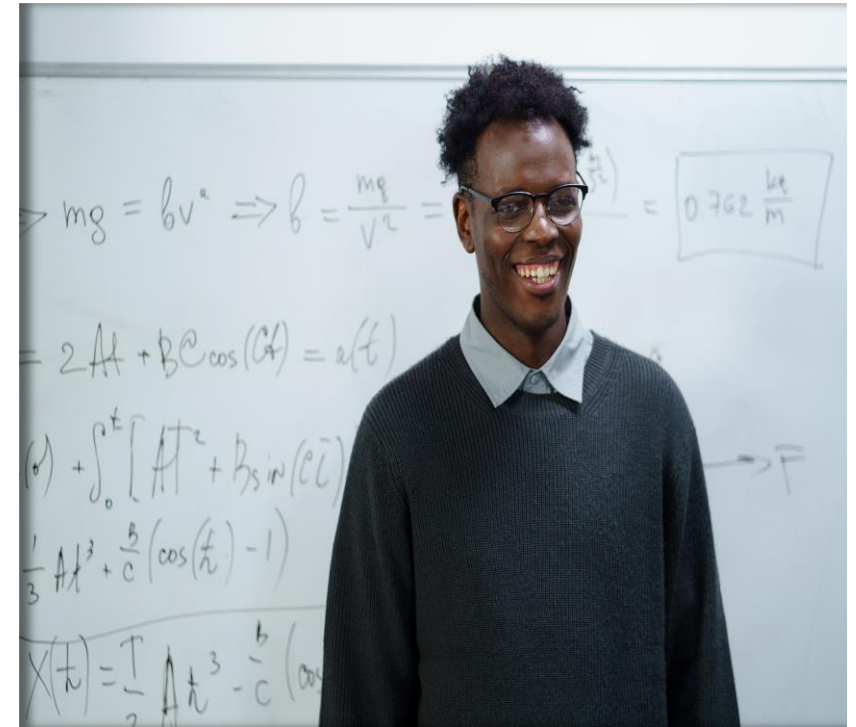
À première vue, comme une image est constituée de pixels et que chaque pixel peut être représenté par un 5-uplet, nous pouvons être tentés, en notant n le nombre de pixels de notre image, de constituer une matrice de n colonnes et 5 lignes. Cependant, cette approche brute ne suffit pas pour des tâches complexes de reconnaissance d'images. Pourquoi ?



La reconnaissance d'image

Extraction de caractéristiques d'une image

Trop de Données à Traiter : Si nous essayons de passer cette matrice directement à des algorithmes de classification comme les machines à vecteurs de support (SVM), nous nous heurterons rapidement à des problèmes de scalabilité. Considérons une image de 1000x1000 pixels, ce qui donne un million de pixels. La matrice de caractéristiques aurait alors un million de colonnes et 5 lignes. La manipulation et le traitement de ces données requièrent des capacités de calcul énormes que même les ordinateurs les plus puissants peinent à gérer efficacement. L'entraînement d'un modèle SVM avec une telle quantité de données serait extrêmement lent et coûteux en ressources.

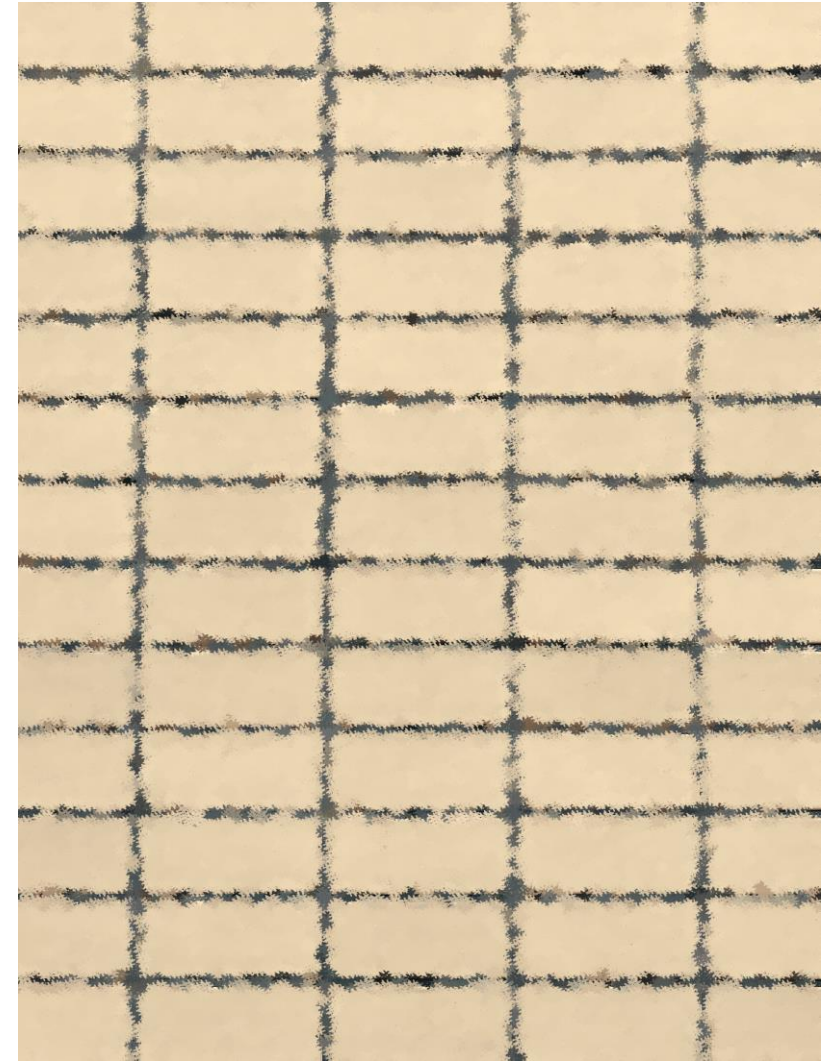


5881 pixel de largeur et 3921 pixel de longueur

La reconnaissance d'image

Extraction de caractéristiques d'une image

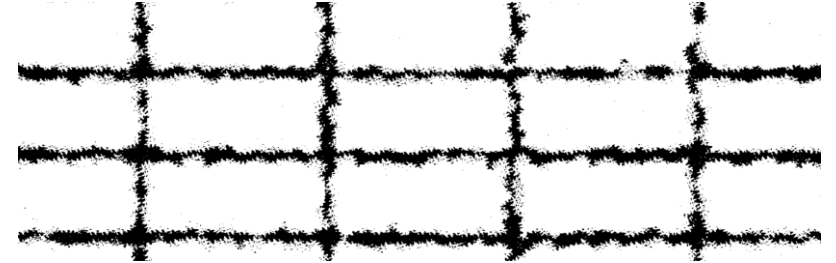
Caractéristiques non informatives : Si, en revanche, nous décidons de simplifier excessivement notre approche en extrayant seulement quelques caractéristiques basiques, comme la longueur et la largeur de l'image ainsi que la couleur du pixel aux quatre coins de l'image, nous obtiendrons une représentation trop pauvre de l'image. Avec si peu d'informations, il serait impossible pour des algorithmes comme les SVM de bien comprendre et de classifier les images. Par exemple, deux images de dimensions identiques et ayant des couleurs similaires aux coins pourraient représenter des objets complètement différents.



La reconnaissance d'image

Extraction de caractéristiques d'une image

Besoin d'Approches Sophistiquées: Pour surmonter ces limitations, il est essentiel d'extraire des caractéristiques plus pertinentes et structurées.



Une approche efficace consiste à utiliser des réseaux de neurones convolutifs pré entraîné tels que VGG16.

La reconnaissance d'image

VGG16 : Un Modèle Puissant pour l'Extraction de Caractéristiques

Comment Fonctionne VGG16 ?



La reconnaissance d'image

VGG16 : Un Modèle Puissant pour l'Extraction de Caractéristiques

Comment Fonctionne VGG16 ?

Vaste sujet ! Retenons juste que c'est un algorithme pré-entraîné sur des millions d'images annotées, capable de faire de la reconnaissance d'images en plus de l'extraction de caractéristiques. Il applique plusieurs filtres à notre image matricielle pour extraire des caractéristiques. Les premières couches détectent des motifs simples comme les bords et les textures, tandis que les couches plus profondes capturent des caractéristiques plus complexes telles que des formes et des objets. Ces caractéristiques extraites sont ensuite utilisées pour classer l'image.

La reconnaissance d'image

VGG16 : Un Modèle Puissant pour l'Extraction de Caractéristiques

Comment Fonctionne VGG16 ?

Le point focal de VGG16 repose sur l'application de différents filtres à la matrice image ainsi que sur le principe du transfert learning.

Un filtre (ou noyau) est une petite matrice de nombres, par exemple une matrice 3x3, utilisée pour détecter des motifs spécifiques dans l'image. Lorsqu'on applique ce filtre à une image de taille 224x224 pixels, il se déplace sur l'image en effectuant une opération de convolution : les valeurs du filtre sont multipliées par les valeurs des pixels sous le filtre, puis additionnées pour produire une nouvelle valeur. Cette valeur est ensuite placée dans une carte de caractéristiques, qui met en évidence les motifs détectés, comme des bords ou des textures. En utilisant plusieurs filtres, VGG16 peut extraire différentes sortes de caractéristiques de l'image, comme des formes et des objets, qui sont ensuite utilisées pour comprendre et classifier l'image.

La reconnaissance d'image

VGG16 : Un Modèle Puissant pour l'Extraction de Caractéristiques

Comment Fonctionne VGG16 ?

Le point focal de VGG16 repose sur l'application de différents filtres à la matrice image ainsi que sur le principe du transfert learning.

Le transfert learning est un principe fondamental qui permet de tirer parti des connaissances acquises par le modèle sur une tâche pour les appliquer à une tâche différente mais similaire. Dans le cas de VGG16, le modèle est pré-entraîné sur un grand ensemble d'images (ImageNet). Les couches initiales du réseau apprennent à détecter des caractéristiques générales qui sont utiles pour une large gamme de tâches de reconnaissance d'images. Lorsqu'on adapte VGG16 à une nouvelle tâche, on peut utiliser ces caractéristiques pré-apprises comme points de départ, en réutilisant les couches convolutionnelles du modèle et en ajustant seulement les couches finales pour la nouvelle tâche. Cela permet d'obtenir des performances élevées même avec des ensembles de données plus petits pour la tâche spécifique, tout en réduisant le temps et les ressources nécessaires pour entraîner le modèle à partir de zéro.

Presentation de SVM

Un problème de classification

Avant toute chose, nous allons commencer par établir un cadre à ce que nous allons voir. En particulier, **qu'est-ce qu'un problème de classification ?**

Considérons l'exemple suivant. On se place dans le plan, et l'on dispose de deux catégories : les ronds rouges et les carrés bleus, chacune occupant une région différente du plan. Cependant, la frontière entre ces deux régions n'est pas connue. Ce que l'on veut, c'est que quand on lui présentera un nouveau point dont on ne connaît que la position dans le plan, l'algorithme de classification sera capable de prédire si ce nouveau point est un carré rouge ou un rond bleu.

Voici notre problème de classification : pour chaque nouvelle entrée, être capable de déterminer à quelle catégorie cette entrée appartient.

Autrement dit, il faut être capable de trouver la frontière entre les différentes catégories. Si on connaît la frontière, savoir de quel côté de la frontière appartient le point, et donc à quelle catégorie il appartient.

Presentation de SVM

Un problème de classification

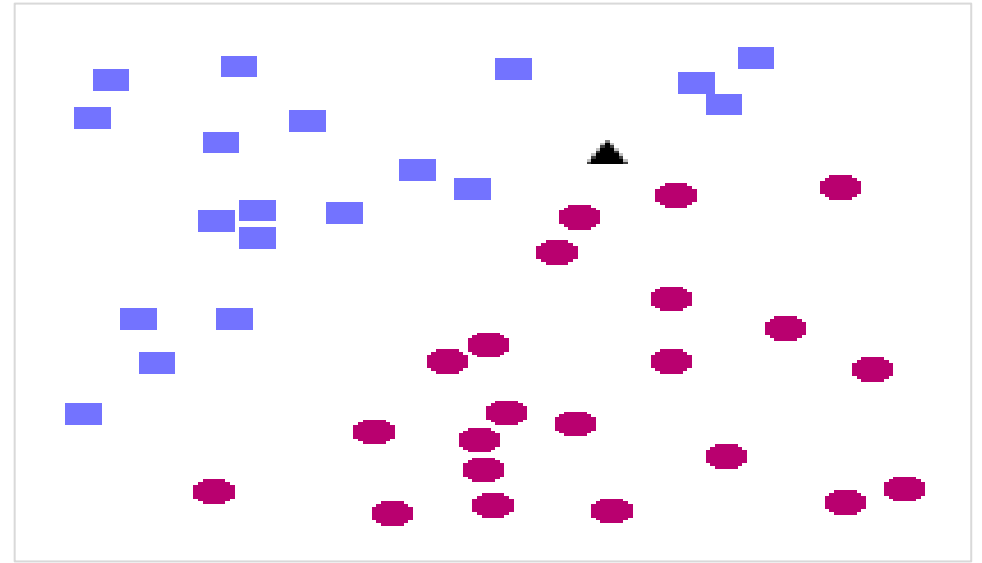


Figure 1. L'objet à classer (le triangle noir) est-il un rond rouge ou bien un carré bleu ?

Le SVM est une solution à ce problème de classification. Le SVM appartient à la catégorie des *classificateurs linéaires* (qui utilisent une séparation linéaire des données), et qui permet de trouver une frontière entre les catégories.

Pour que le SVM puisse trouver cette frontière, il est nécessaire de lui donner **des données d'entraînement**. En l'occurrence, on donne au SVM un ensemble de points, dont on sait déjà si ce sont des carrés rouges ou des ronds bleus, comme dans la Figure 1. A partir de ces données, le SVM va estimer l'emplacement le plus plausible de la frontière : c'est la période d'**entraînement**, nécessaire à tout algorithme d'apprentissage automatique.

Presentation de SVM

Un problème de classification

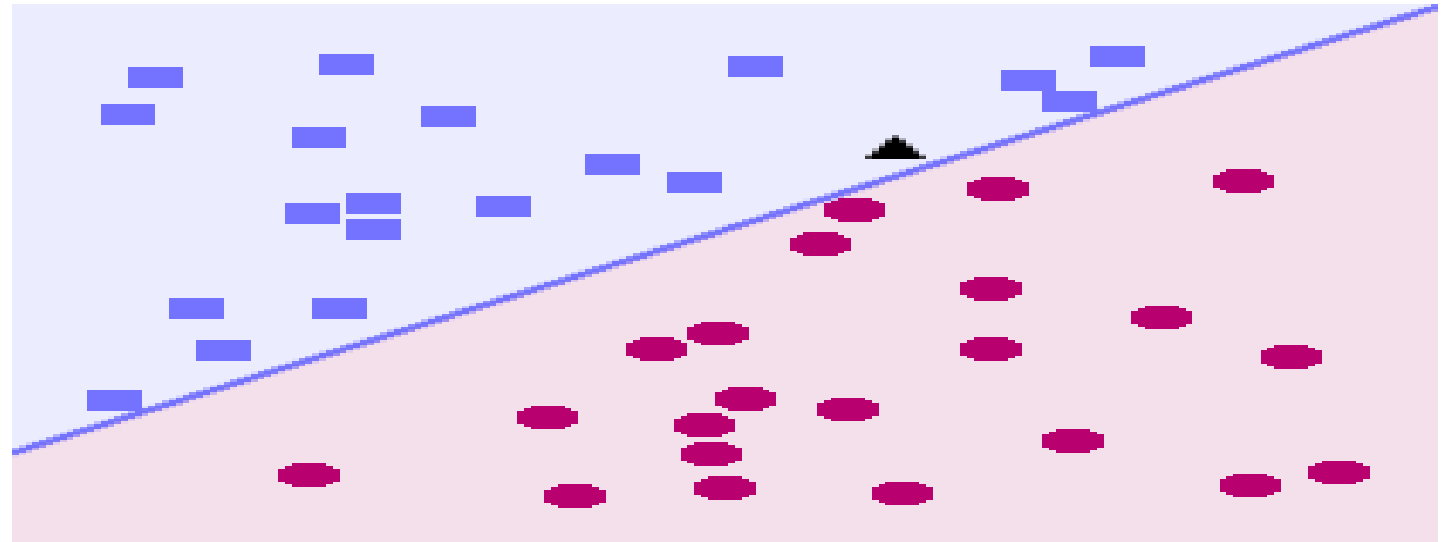


Figure 2. Notre SVM, muni des données d'entraînement (les carrés bleus et les ronds rouges déjà indiqués comme tels par l'utilisateur), a tranché : le triangle noir est en fait un carré bleu.

Une fois la phase d'entraînement terminée, le SVM a ainsi trouvé, à partir de données d'entraînement, l'emplacement supposé de la frontière. En quelque sorte, il a « appris » l'emplacement de la frontière grâce aux données d'entraînement. Qui plus est, le SVM est maintenant **capable de prédire** à quelle catégorie appartient une entrée qu'il n'avait jamais vue avant, et sans intervention humaine (comme c'est le cas avec le triangle noir dans la Figure 2) : c'est là tout l'intérêt de l'apprentissage automatique.

Presentation de SVM

Les SVM dans les grandes lignes

Bon, nous savons donc que le but, pour un SVM, est d'apprendre à bien placer la frontière entre deux catégories. Mais comment faire ? Quand on a un ensemble de points d'entraînement, il existe plusieurs lignes droites qui peuvent séparer nos catégories. La plupart du temps, il y en a une infinité... Alors, laquelle choisir ?

Intuitivement, on se dit que si on nous donne un nouveau point, très proche des ronds rouges, alors ce point a de fortes chances d'être un rond rouge lui aussi. Inversement, plus un point est près des carrés bleus, plus il a de chances d'être lui-même un carré bleu. Pour cette raison, un SVM va placer la frontière aussi loin que possible des carrés bleus, mais également aussi loin que possible des ronds rouges.

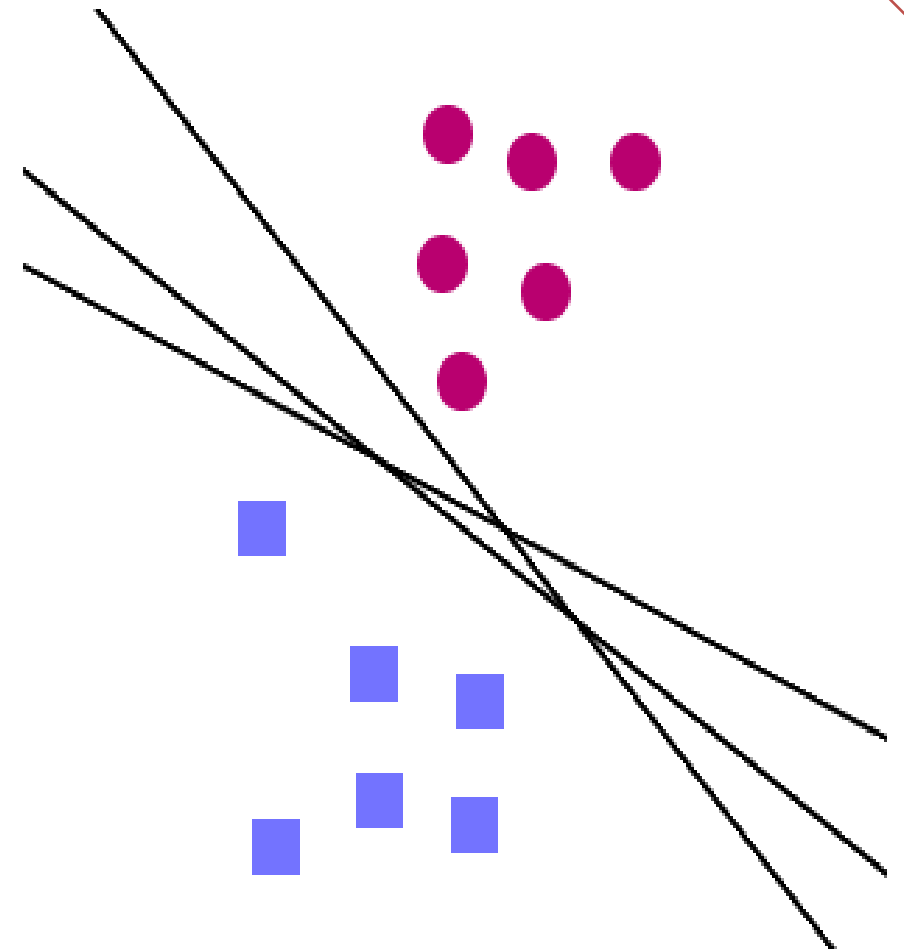


Figure 3 : Pour un ensemble de données d'entraînement, il existe plusieurs frontières possibles

Presentation de SVM

Les SVM dans les grandes lignes

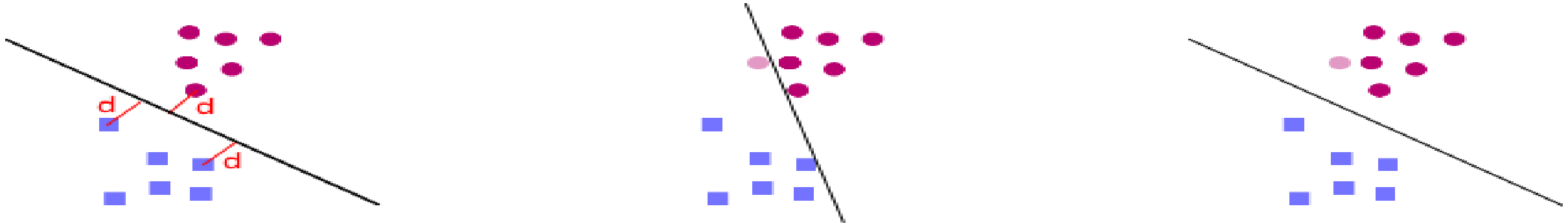


Figure 4 : A gauche, on maximise la distance d entre la frontière et les points d'entraînement. Au centre, une frontière non-optimale, qui passe très près des points d'entraînement. Cette frontière classe de façon erronée le rond rouge clair comme un carré bleu ! Au contraire, à droite, la frontière optimale (qui maximise d) classe bien le rond rouge clair comme un rond rouge.

Comme on le voit dans la figure 4, c'est bien la frontière la plus éloignée de tous les points d'entraînement qui est optimale, on dit qu'elle a la meilleure **capacité de généralisation**. Ainsi, le but d'un SVM est de trouver cette frontière optimale, en maximisant la distance entre les points d'entraînement et la frontière.

Les points d'entraînement les plus proches de la frontière sont appelés vecteurs support

Présentation de SVM

Cas de donnée non linéairement separable ...

Considérons l'exemple suivant (figure 5). Puisque les carrés sont entourés de ronds de toute part, il est impossible de trouver de ligne droite qui soit une frontière : on dit que les données d'entraînement ne sont pas linéairement séparables. Cependant, imaginez qu'on arrive à trouver une transformation qui fasse en sorte que notre problème ressemble à ça (figure 6):



Figure 6 : Les mêmes points d'entraînement, après transformation

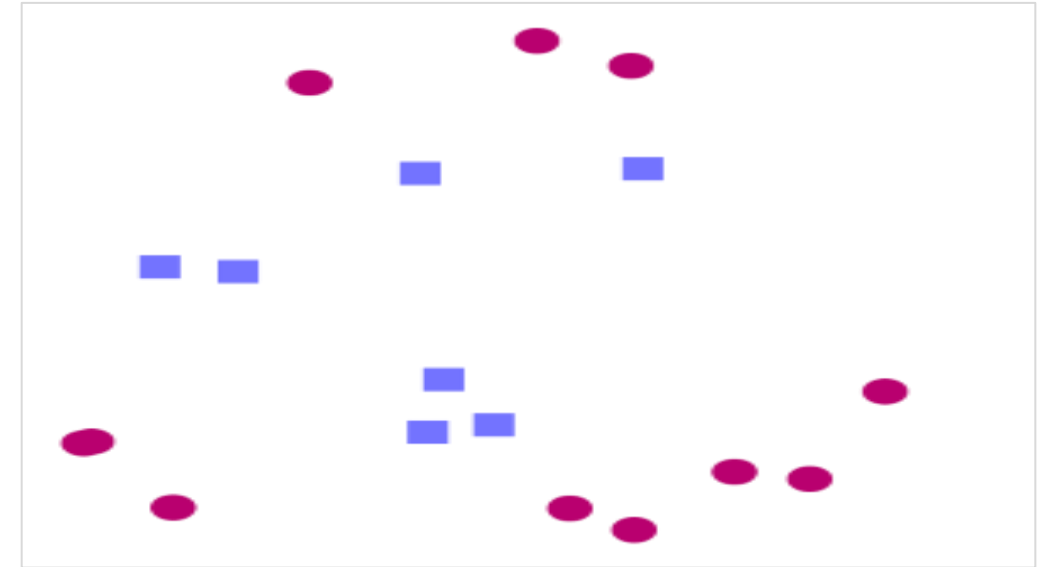


Figure 5 : Comment séparer les deux catégories avec une ligne droite ?

A partir de là, il est facile de trouver une séparation linéaire. Il suffit donc de trouver une transformation qui va bien pour pouvoir classer les objets. Cette méthode est appelée kernel trick (astuce du noyau). Vous vous en doutez, toute la difficulté est de trouver la bonne transformation...

3

Formulation mathématique du problème et cas pratique

Formulation mathématique et présentations
d'un cas pratique sous python.

Formulation mathématique

Espace vectoriel et hyper-plan

- ❑ De façon plus générale que dans les exemples donnés précédemment, les SVM ne se bornent pas à séparer des points dans le plan. Ils peuvent en fait séparer des points dans un espace de dimension quelconque.
- ❑ Fondamentalement, un SVM cherchera simplement à trouver un hyperplan qui sépare les deux catégories de notre problème.
- ❑ Dans un espace vectoriel de dimension finie n , un hyperplan est un sous-espace vectoriel de dimension $n - 1$. Ainsi, dans un espace de dimension 2 un hyperplan sera une droite, dans un espace de dimension 3 un hyperplan sera un plan, etc.

Formulation mathématique

Espace vectorielle et hyperplan

Précisions sur les hyperplans

- ❑ Soit E un espace vectoriel de dimension n . L'équation caractéristique d'un hyperplan est de la forme $w_1x_1 + w_2x_2 + \dots + w_nx_n = 0$ (*) ou w_1, \dots, w_n sont des scalaires. Par définition pour

tout vecteur $x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in E$ vérifiant l'équation (*) appartient à l'hyperplan.

- ❑ De plus, un hyperplan sépare complètement l'espace vectoriel en deux parties distinctes. Ainsi, une ligne droite sépare le plan en deux régions distinctes ; et en dimension 3, c'est le plan qui joue le rôle de ce séparateur : il y a un demi-espace « d'un côté » du plan, et un autre demi-espace « de l'autre côté » du plan. Avec un hyperplan, il est donc possible de diviser notre espace vectoriel en deux catégories distinctes : **tout pile ce qu'il nous faut pour notre problème de classification !**



Formulation mathématique

Espace vectorielle et hyperplan

- ❑ un hyperplan vectoriel passe toujours par l'origine pour cette raison nous utiliserons des hyperplan affine (cf. Geometrie affine) 🍊
- ❑ Pendant son entraînement le SVM calculera un hyperplan vectoriel d'équation caractérisé par le vecteur w et le scalaire b et vérifiant $w_1x_1 + w_2x_2 + \dots + w_nx_n + b = 0$
- ❑ Le vecteur $w = (w_1, \dots, w_n)^T$ est appelé vecteur poids et le scalaire b est le biais.
- ❑ Une fois l'entraînement terminé, pour classer une nouvelle entrée $x = (a_1, \dots, a_n)^T$, le SVM regarde le signe de $h(x) = w_1a_1 + w_2a_2 + \dots + w_na_n + b = w^T \cdot x + b$
- ❑ Si $h(x)$ est positif ou nul, alors x est d'un côté de l'hyperplan affine et appartient à la première catégorie, sinon x est de l'autre côté de l'hyperplan, et donc appartient à la seconde catégorie.
- ❑ En résumé, on souhaite savoir, pour un point x , s'il se trouve d'un côté ou de l'autre de l'hyperplan. La fonction h nous permet de répondre à cette question, grâce à la classification

suivante:
$$\begin{cases} h(x) \geq 0 \Rightarrow x \in \text{catégorie 1} \\ h(x) < 0 \Rightarrow x \in \text{catégorie 2} \end{cases}$$



Formulation mathématique

Entraînement des SVM

- ❑ Supposons que l'on assigne à tout point $x_k \in \mathbb{R}^n$ un label l_k qui vaut 1 si x_k appartient à la première catégorie, et -1 si x_k appartient à la seconde catégorie. Alors, si le SVM est correctement entraîné, on a toujours $l_k h(x_k) \geq 0$, c'est-à-dire $l_k(w^T \cdot x_k + b) \geq 0$
- ❑ Le but d'un SVM, lors de l'entraînement, est donc de trouver un vecteur de poids w et un biais b tels que, pour tout point x_k de label l_k appartenant aux données d'entraînement, $l_k(w^T \cdot x_k + b) \geq 0$.
- ❑ Autrement dit, de trouver un hyperplan séparateur entre les deux catégories.
- ❑ on choisira l'hyperplan qui maximise la marge, c'est-à-dire la distance minimale entre les vecteurs d'entraînement et l'hyperplan. De tels vecteurs situés à la distance minimale sont appelés vecteurs supports.
- ❑ Ce choix est d'ailleurs un résultat mathématiquement fonde

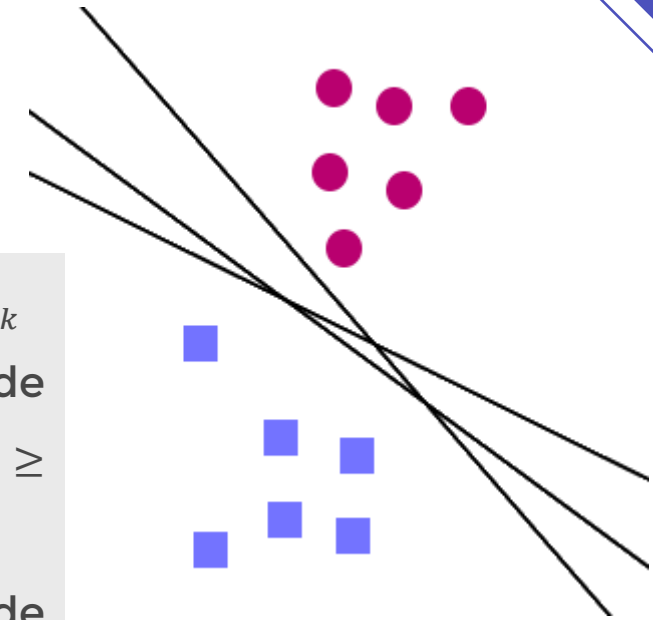


Figure 7 : Quel hyperplan choisir ?

Formulation mathématique

Normalisation et résolution du problème

- ❑ Pour un vecteur x_k sa distance à l'hyperplan de vecteur support w et de biais b est donnée par : $\frac{l_k(w^T \cdot x_k + b)}{\|w\|}$ ou $\|w\|$ désigne la norme euclidienne de w .
- ❑ On appelle marge la distance minimale entre l'hyperplan à un des points d'entraînement.
- ❑ La marge d'un hyperplan de paramètre (w, b) par rapport à un ensemble de points x_k est $\min_k \frac{l_k(w^T \cdot x_k + b)}{\|w\|}$
- ❑ On veut trouver l'hyperplan de support w et de biais b qui permet de maximiser cette marge. Cela permettra, intuitivement, d'être tolérant face aux petites variations.



Figure 8 : La capacité de généralisation est meilleure à droite (avec l'hyperplan optimal) qu'à gauche

Formulation mathématique

Normalisation et résolution du problème

Ainsi notre problème consiste à chercher **l'unique** hyperplan dont les paramètres (w, b) sont donnés par la formule:

$$\max_{w, b} \min_k \frac{l_k(w^T \cdot x_k + b)}{\|w\|}$$

* L'unicité est démontrée par Vapnik et Kotz en 1982



Formulation mathématique

Normalisation et résolution du problème

- ❑ Mathématiquement il est évident qu' en multipliant le vecteur w et le scalaire b par un même scalaire non nul, on obtient le même hyperplan affine, donc plusieurs couples de paramètres (w,b) peuvent représenter un même hyperplan affine.
- ❑ **Rappels:** les vecteurs supports sont les points les plus proches de l'hyperplan séparateur.
- ❑ **Rappels:** On travaille avec un nombre fini de vecteurs
- ❑ Pour un hyperplan (w,b) on décide de ne considérer l'unique paramétrage (w,b) tel que les vecteurs supports x_s vérifient $l_s(w^T \cdot x_s + b) = 1$ (il suffit de multiplier nos w et b initiaux par un scalaire $\frac{1}{v}$ tel que $l_s(w^T \cdot x_s + b) = v$)
- ❑ Par conséquent, $\forall k \quad l_k(w^T \cdot x_k + b) \geq 1$ ou x_k est un point quelconque d'entraînement et l'égalité est atteinte si x_k est un vecteur support.
- ❑ Dit autrement, cette formulation permet de garantir que la marge $\min_k \frac{l_k(w^T \cdot x_k + b)}{\|w\|}$ est alors $\frac{1}{\|w\|}$
- ❑ Le problème d'optimisation de la marge se simplifie donc en $\max_{w,b} \frac{1}{\|w\|}$ avec (w,b) sous hypothèse de normalisation 🍊

Formulation mathématique

Normalisation et résolution du problème

On se retrouve avec le probleme suivant :

$$\left\{ \begin{array}{l} \text{Maximiser } \frac{1}{\|w\|} \\ \text{Sous contrainte, } \forall k \quad l_k(w^T \cdot x_k + b) \geq 1 \end{array} \right.$$

Que l'on reformule en

$$\left\{ \begin{array}{l} \text{Minimiser } \|w\| \\ \text{Sous contrainte, } \forall k \quad l_k(w^T \cdot x_k + b) \geq 1 \end{array} \right.$$

Que pour des raisons pratique,on reformule en

$$\left\{ \begin{array}{l} \text{Minimiser } \frac{\|w\|^2}{2} \\ \text{Sous contrainte, } \forall k \quad l_k(w^T \cdot x_k + b) \geq 1 \end{array} \right.$$

Formulation mathématique

Normalisation et résolution du problème

$$\left\{ \begin{array}{l} \text{Minimiser } \frac{\|w\|^2}{2} \\ \text{Sous contrainte, } \forall k \in [1; p] \quad l_k(w^T \cdot x_k + b) \geq 1 \end{array} \right.$$

* On suppose avoir p points dans la base d'entraînement

Ce genre de problème est appelé problème d'optimisation quadratique, dans le cas présent on utilisera la méthode du multiplicateur de Lagrange pour le résoudre.

Le lagrangien est donnée par $L(w, b, \lambda) = \frac{1}{2} \|w\|^2 - \sum_{k=1}^{k=p} \lambda_k [l_k(w^T \cdot x_k + b) - 1]$ (1)

En annulant les dérivé partielle selon les conditions de Kuhn-Tucker, on obtient :

$\sum_{k=1}^{k=p} \lambda_k l_k x_k = w^*$ et $\sum_{k=1}^{k=p} \lambda_k l_k = 0$. En reinjectant ces valeurs dans l'equation (1), on obtient la formulation duale:

Maximiser $\tilde{L}(\lambda) = \sum_{k=1}^{k=p} \lambda_k - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j l_i l_j x_i^T x_j$ **sous les contraintes** $\forall k \in [1; p] \quad \lambda_k \geq 0$ et $\sum_{k=1}^{k=p} \lambda_k l_k = 0$

Formulation mathématique

Normalisation et résolution du problème

Démonstration:

D'abord on calcul b sachant : $l_s(w^T \cdot x_s + b) = 1$ en remplaçant w par sa valeur optimal on trouve b .

Puis on remplace w et b dans l'équation (1). Après simplification, On obtient: $\tilde{L}(\lambda) = \sum_{k=1}^{k=p} \lambda_k - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j l_i l_j x_i^T x_j$

La contrainte $\sum_{k=1}^{k=p} \lambda_k l_k = 0$ proviens du fait que la dérivé partielle de L par rapport a b est nulle. cqfd

Maximiser $\tilde{L}(\lambda) = \sum_{k=1}^{k=p} \lambda_k - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j l_i l_j x_i^T x_j$ sous les contraintes $\forall k \in [1; p] \lambda_k \geq 0$ et

Formulation mathématique

Normalisation et résolution du problème

* On suppose avoir p points dans la base d'entraînement

Ce qui donne les multiplicateurs de Lagrange optimaux λ_k^* .

Afin d'obtenir l'hyperplan solution, on remplace w par sa valeur optimale w^* , dans l'équation de l'hyperplan $h(x)$, ce qui donne :

$$h(x) = w^{*T} \cdot x + b \text{ avec } w^* = \sum_{k=1}^{k=p} \lambda_k^* l_k x_k$$

Remarque:

□ La contrainte d'exclusion de Kuhn & Tucker donne: $\lambda_k[l_k h(x_k) - 1] = 0$ ce qui implique que Les seuls points pour lesquels les contraintes du lagrangien sont actives sont donc les vecteurs supports. En d'autres termes, seuls les vecteurs supports participent à la définition de l'hyperplan optimal.

Formulation mathématique

Normalisation et résolution du problème

* On suppose avoir p points dans la base d'entraînement

Maximiser $\tilde{L}(\lambda) = \sum_{k=1}^{k=p} \lambda_k - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j l_i l_j x_i^T x_j$ sous les contraintes $\forall k \in [1;p] \lambda_k \geq 0$ et $\sum_{k=1}^{k=p} \lambda_k l_k = 0$

Remarque:

- l'hyperplan solution ne dépend que du produit scalaire entre le vecteur d'entrée et les vecteurs supports.
- le fait d'ajouter des échantillons à l'ensemble d'apprentissage qui ne sont pas des vecteurs supports n'a aucune influence sur la solution finale.

Formulation mathématique

Hyperparamètre et marge souple

En réalité, trouver un hyperplan séparateur parfait n'est pas toujours possible, et même s'il existe, il ne représente pas nécessairement la meilleure solution pour la classification. De plus, une erreur d'étiquetage dans les données d'entraînement (par exemple, un exemple étiqueté +1 au lieu de -1) peut affecter considérablement l'hyperplan.

Dans le cas où les données ne sont pas linéairement séparables ou contiennent du bruit (comme des données mal étiquetées), les contraintes du problème d'optimisation quadratique des SVM ne peuvent pas être toujours vérifiées, et il devient nécessaire de les relaxer. C'est là qu'intervient la notion de marge souple.

La marge souple permet d'introduire une certaine tolérance aux erreurs de classification dans le processus d'apprentissage des SVM. Cela se traduit par l'introduction de variables de relaxation ("slack variables") dans le problème. Ces variables permettent à certains points de données de se situer du mauvais côté de l'hyperplan séparateur, tant que cela n'excède pas une certaine marge définie par le paramètre de régularisation C .

En permettant une certaine marge d'erreur, les SVM avec marge souple sont moins sensibles aux erreurs dans les données d'entraînement et peuvent mieux gérer les données bruitées ou non linéairement séparables. **Pour ce faire, nous introduisons des variables ε_i sur les contraintes, représentant la distance à laquelle un point est éloigné d'un bon classement, puis minimiser la somme des ε_i . Les variables ε_i sont appelées variables de relaxation. La contrainte de notre problème initial devient :**

Formulation mathématique

Hyperparamètre et marge souple

La contrainte de notre problème initial devient : $\forall k \ l_k(w^T \cdot x_k + b) \geq 1 - \varepsilon_k$.

Si $\varepsilon_k = 0$ alors x_k est sur la marge. Sinon x_k est mal positionné et ε_k représente la distance de violation de la marge par ce point.

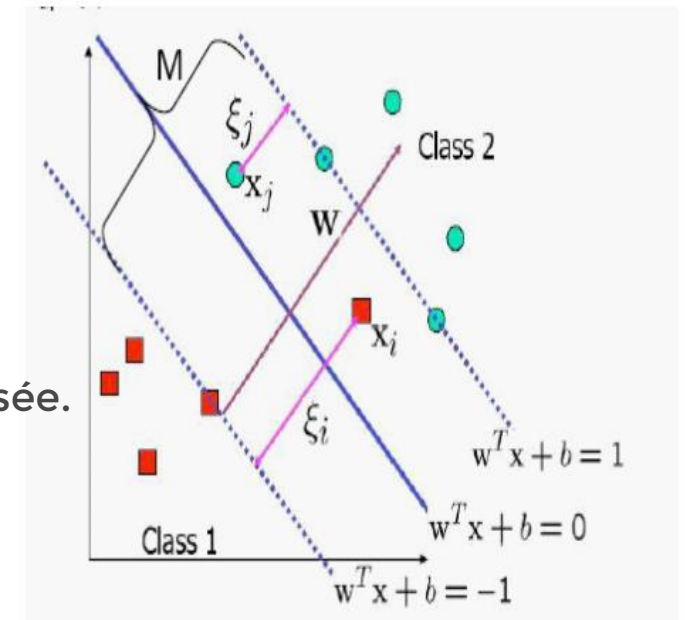
Au lieu de rechercher uniquement un hyperplan séparateur qui maximise la marge, on recherche un hyperplan qui minimise aussi la somme des erreurs

Permettes ε_i . Le problème d'optimisation devient :

$$\left\{ \begin{array}{l} \text{Minimiser } \frac{\|w\|^2}{2} + C \sum_{k=1}^{k=p} \varepsilon_i \\ \text{Sous contrainte, } \forall k \ l_k(w^T \cdot x_k + b) \geq 1 - \varepsilon_i \text{ et } \varepsilon_i \geq 0 \end{array} \right.$$

Où $C > 0$ est une constante qui représente la pénalité d'avoir des données mal classées.

Lorsque C est très élevé, il y aura très peu de données mal classées.



Formulation mathématique

Optimisation minimale séquentielle (SMO)

Par analogies au problème initial, le problème dual devint :

Maximiser $\tilde{L}(\lambda) = \sum_{k=1}^{k=p} \lambda_k - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j l_i l_j x_i^T x_j$ **sous les contraintes** $\forall k \in [1; p] \ 0 \leq \lambda_k \leq C$ et $\sum_{k=1}^{k=p} \lambda_k l_k = 0$

Pour résoudre ce problème, nous allons utiliser l'algorithme SMO, elle se résume à :

Répéter jusqu'à convergence :

{

1 Sélectionnez une paire λ_i et λ_j

2 Réoptimisez $\tilde{L}(\lambda)$ par rapport à λ_i et λ_j tout en maintenant tous les autres λ_k .

}

Pour des contraintes de temps, la présentation complète de cet algorithme dépasse le cadre de ce travail. Nous invitons donc ceux qui souhaitent en savoir davantage à se référer au mémoire de Mlle Kaim Houaria, intitulé **Modélisation, Contrôle et Optimisation**, à la page 24.

Formulation mathématique

Optimisation minimale séquentielle (SMO)

Par analogies au probleme initiale, le probleme duale deviant:

Maximiser $\tilde{L}(\lambda) = \sum_{k=1}^{k=p} \lambda_k - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j l_i l_j x_i^T x_j$ **sous les contraintes** $\forall k \in [1; p] \ 0 \leq \lambda_k \leq C$ *et* $\sum_{k=1}^{k=p} \lambda_k l_k = 0$

Après avoir determine les λ_i on calcul w grace aux conditions KKT du probleme duale avec marge souple.

$\sum_{k=1}^{k=p} \lambda_k l_k x_k = w^*$ et pour b , il s'obtient en prenant la moyenne des biais b calculés à chaque étape de la boucle d'optimisation de sorte que les conditions optimales KKT soit verifie a chaque etape.

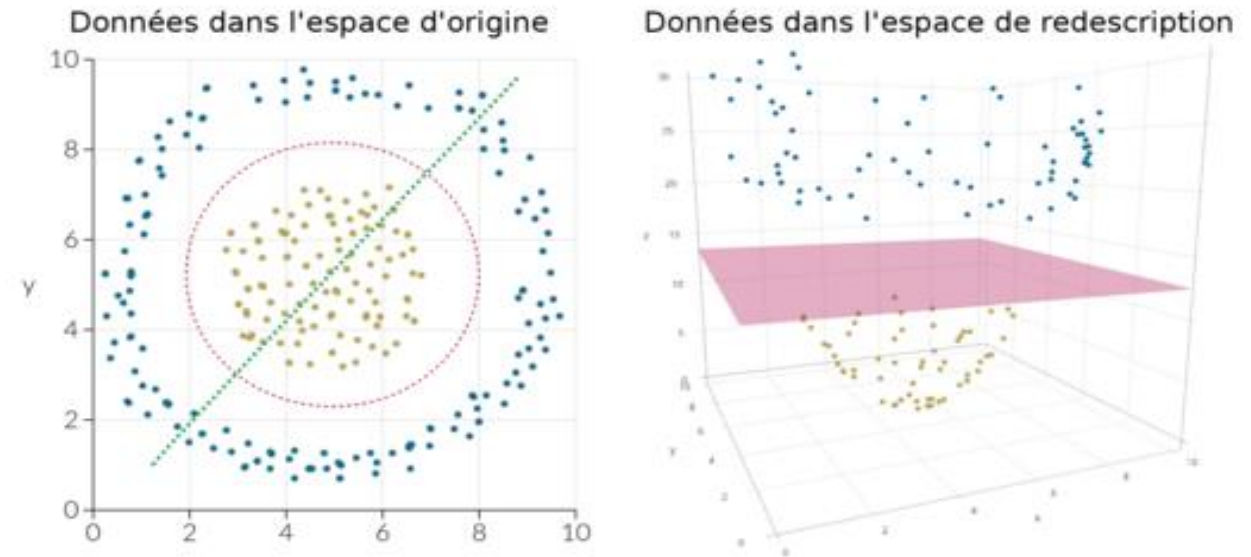
Probleme des donnee non lineairement separable

Astuce du noyau

Figure 9 : Données d'entraînement non linéairement séparables

Pour des jeux de données complexes, il n'existe en général pas de séparatrice linéaire. Imaginons par exemple un plan dans lequel les points en bleu sont regroupés à l'intérieur d'un cercle, avec des points rouges tout autour : aucun séparateur linéaire ne peut correctement séparer les groupes : le problème n'est pas linéairement séparable. Il n'existe pas d'hyperplan séparateur.

Cependant il existe une astuce pour éventuellement tenter de séparer un jeu de données non linéairement séparable : Passer en dimension supérieure grâce à l'astuce du noyau (voir wikipedia pour plus de détails)



Cas pratique sur python

Choix des auteurs

Nous avons décidé de ne pas utiliser VGG16 pour deux raisons :

- **Pré-entraînement** : VGG16 est un modèle pré-entraîné dont les caractéristiques extraites sont déjà suffisamment élaborées pour la plupart des tâches de classification, rendant l'utilisation d'un SVM pour la classification redondante.
- **Besoins pédagogiques** : Pour des raisons pédagogiques, nous souhaitons obtenir des visualisations des caractéristiques. Cependant, VGG16 fournit un vecteur de caractéristiques de grande taille, ce qui rend difficile l'interprétation et la visualisation des caractéristiques extraites.



Nous avons opté pour une extraction manuelle ; par exemple, pour des images de fleurs, nous pouvons nous intéresser à la longueur du pétale, à la largeur du pétale, ainsi qu'à la longueur et à la largeur du sépale. Justement, il existe une base de données appelée Iris qui fournit ces mesures pour différents échantillons de fleurs.

Cas pratique sur python

Presentation des fleurs d'iris

Ce jeu de données comprend un total de 150 observations, réparties de manière égale entre les trois espèces de fleurs d'iris (setosa, virginica et versicolor). Quatre caractéristiques sont mesurées pour chaque observation (c.-à-d. la longueur et la largeur du sépale et du pétale, en centimètres).

iris setosa



petal

sepal

iris versicolor



petal

sepal

iris virginica



petal

sepal

Les auteurs du projet



HANNA

Elève ingénieur
statisticienne économiste

Econimiste



Sadibou

Elève ingénieur
statisticien économiste

Mathématicien



Moussa

Elève ingénieur
statisticien économiste

Mathematicien



Merci de votre attention