



**CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS**  
**CENTRE RÉGIONAL ASSOCIÉ DE NOUVELLE-AQUITAINE**

---

**Mémoire probatoire présenté en vue d'obtenir**  
**UE « Information et communication pour ingénieur »**  
**Spécialité :**  
**Informatique, Réseaux, Systèmes et Multimédia**

**par**

**Marc BENINCA**

---

**SVM**  
**Machine à Vecteurs de Support**

**Soutenu le 16 Septembre 2020**

---

**JURY**

<b>PRÉSIDENT :</b>	<b>Civilité</b>	<b>Prénom</b>	<b>NOM</b>	<b>Fonction</b>	<b>Organisme</b>
<b>MEMBRES :</b>	<b>Civilité</b>	<b>Prénom</b>	<b>NOM</b>	<b>Fonction</b>	<b>Organisme</b>
	<b>Civilité</b>	<b>Prénom</b>	<b>NOM</b>	<b>Fonction</b>	<b>Organisme</b>

# Abréviations

**ANOVA** ANalysis Of VAriance. [14](#)

**HBOS** Histogram Based Outlier Score. [5](#)

**IRM** Imagerie par Résonance Magnétique. [14](#)

**NN** Nearest Neighbors. [5](#)

**RBF** Radial Basis Function. [6](#), [12-14](#)

**SGBD** Systèmes de Gestion de Bases de Données. [4](#)

**SI** Systèmes d'Information. [4](#), [6](#)

**SVC** Support Vector Classification. [6](#)

**SVM** Support Vector Machine. [5](#), [6](#), [14](#), [20](#), [22](#), [24](#)

**SVR** Support Vector Regression. [6](#)

**TIC** Technologies d'Information et de Communication. [4](#)

# Glossaire

**big data** données massives. [4](#), [24](#)

**clustering** regroupement d'éléments en sous-ensembles caractéristiques. [4](#), [5](#)

**dataset** ensemble de données. [4](#), [6](#), [12](#), [14](#), [15](#), [17](#), [18](#), [20](#), [24](#)

**fonction noyau** transformation non linéaire permettant une séparation linéaire. [6](#), [12](#), [13](#), [20-22](#)

**grid search** algorithme d'affinage d'hyperparamètres par grille de valeurs. [12](#), [21](#)

**hyperplan** sous-espace en  $n-1$  dimensions d'un espace en  $n$  dimensions. [6](#), [18](#), [20](#)

**kernel trick** astuce du noyau pour éviter des calculs plus complexes. [12](#), [13](#), [22](#)

**machine learning** apprentissage machine automatique. [4](#), [15](#), [22](#), [24](#)

**sigmoïde** courbe à double asymptôte 0 et 1, centrée sur  $1/2$ . [6](#), [14](#)

**stéganographie** dissimulation d'informations dans un plus grand ensemble. [6](#)

**support vector** sous-ensemble de données d'entraînement. [8](#), [9](#), [15](#), [20](#), [21](#)



# SVM

## Machine à Vecteurs de Support

### Plan

<b>Abréviations</b> . . . . .	<b>1</b>
<b>Glossaire</b> . . . . .	<b>2</b>
<b>Plan</b> . . . . .	<b>3</b>
<b>1 Introduction</b> . . . . .	<b>4</b>
1.1 Machine learning . . . . .	4
1.2 Méthodes . . . . .	4
1.2.1 Méthodes non supervisées . . . . .	4
1.2.2 Méthodes semi-supervisées . . . . .	5
1.2.3 Méthodes supervisées . . . . .	5
<b>2 Principes</b> . . . . .	<b>6</b>
2.1 Régression . . . . .	7
2.1.1 Régression linéaire . . . . .	7
2.1.2 Régression non linéaire . . . . .	7
2.2 Classification . . . . .	8
2.2.1 Classification linéaire . . . . .	8
2.2.2 Classification non linéaire . . . . .	10
2.2.3 Classification multi-classes . . . . .	13
2.2.4 Optimisation . . . . .	13
<b>3 Exemples</b> . . . . .	<b>14</b>
3.1 Linéaire . . . . .	14
3.2 Non linéaire . . . . .	16
<b>4 Critiques</b> . . . . .	<b>18</b>
4.1 Avantages . . . . .	18
4.2 Inconvénients . . . . .	19
4.3 Limitations . . . . .	19
<b>5 Conclusion</b> . . . . .	<b>20</b>
<b>Références</b> . . . . .	<b>21</b>

**Figures** . . . . . 22

**Résumé** . . . . . 23

# 1 Introduction

Les [Technologies d'Information et de Communication \(TIC\)](#) ont fait émerger des [Systèmes d'Information \(SI\)](#) de grande envergure, qui génèrent continuellement une quantité globale toujours plus importante de données, communément appelée [big data](#).

Les volumes démesurés de données manipulés par ces [SI](#) sont clairement incompatibles avec les principes classiques des [Systèmes de Gestion de Bases de Données \(SGBD\)](#). De nouvelles méthodologies ont donc dû être créées afin de pouvoir analyser puis exploiter ces [big data](#).

## 1.1 Machine learning

Le [machine learning](#) regroupe les différentes nouvelles approches méthodologiques permettant de faire ressortir une compréhension des [big data](#), en extraire des [datasets](#) exploitables puis, à partir de cet apprentissage, être en mesure de comprendre de nouvelles données.

Les 2 défis à relever pour être pertinents dans cette démarche :

- ne pas surentraîner (spécialiser) l'apprentisseur automatique, pour conserver une capacité de généralisation
- pour chaque type de problèmes à résoudre, choisir les algorithmes les plus pertinents

## 1.2 Méthodes

### 1.2.1 Méthodes non supervisées

Ces premières méthodes permettent de travailler avec un ensemble d'éléments ne disposant pas d'étiquetage préalable, elles mettent ainsi en place des mécanismes permettant de faire du [clustering](#) de ces éléments.

Exemples d'algorithmes :

- k-means (moyennes)
- réduction de dimensionnalité
- réseaux de neurones
- analyse des composants principaux ou indépendants
- modèles de distribution

- classification hiérarchique
- [clustering](#) par décalage moyen
- Apriori

### 1.2.2 Méthodes semi-supervisées

Ces méthodes autorisent l'utilisation d'un ensemble d'éléments hétérogène, certains ayant déjà été étiquetés au préalable, alors que d'autres non. Un mélange de différentes techniques s'avère donc nécessaire.

Exemples d'algorithmes :

- [Histogram Based Outlier Score \(HBOS\)](#)
- forêts isolées
- autoencodeurs

### 1.2.3 Méthodes supervisées

Ces méthodes utilisent 2 sous-ensembles d'éléments intégralement étiquetés, ayant une valeur de sortie pour plusieurs valeurs de variables en entrée. Les données d'apprentissage permettent un entraînement préalable du modèle. Les données de test permettent ensuite d'en évaluer la pertinence.

Le modèle permet enfin, à partir de nouvelles valeurs en entrée, de fournir des valeurs de sortie prédictives de 2 types :

- régression → la valeur est un nombre
- classification → la valeur est une catégorie, parmi un ensemble défini

Exemples d'algorithmes :

- régression linéaire, logistique ou vectorielle
- arbre de régression ou classification
- [k-Nearest Neighbors \(NN\)](#) (plus proches voisins)
- classificateur Naïve Bayes
- réseaux de neurones
- [Support Vector Machine \(SVM\)](#)

Ce sont ces [SVM](#) qui feront l'objet d'étude du présent document. En seront passés en revue les différents principes techniques sous-jacents, deux exemples d'application, ainsi que des critiques sur ce type d'algorithmes.

## 2 Principes

L'approche **SVM** est un ensemble de méthodes supervisées utilisant :

1. un **dataset** d'apprentissage pour entraîner l'algorithme, et qui fait donc office de superviseur
2. un **dataset** de test pour vérifier sa pertinence

Cette approche se révèle appropriée dans de nombreux cas d'utilisation :

- filtrage d'email, courriel légitime ou pourriel (phishing, spam)
- classification d'images, quel que soit le **SI**
- détection de **stéganographie** dans des fichiers multimédias
- quantification de granularité dans des textures
- reconnaissance de caractères et d'écriture dans des images
- classification d'expressions faciales dans des images
- reconnaissance vocale dans des échantillons sonores
- classification et prédiction de structure de protéines
- établissement de diagnostics médicaux
- classification de documents texte en différentes catégories

En fonction du type de problèmes, deux types de résolution :

- **régression** (**Support Vector Regression (SVR)**) → nombre
- **classification** (**Support Vector Classification (SVC)**) → catégorie

En fonction des **datasets**, deux types d'approche mathématique :

- **linéaire** : la plus simple
- **non linéaire** : faisant appel à des **fonctions noyau**

Quatre paramètres permettent d'affiner le modèle :

- **noyau** : linéaire, **Radial Basis Function (RBF)**, polynomial, **sigmoïde**, etc.
- **degré** : aide à trouver un **hyperplan** séparateur en contexte polynomial, faisant rapidement augmenter le temps nécessaire à l'entraînement
- **$\gamma$**  : pour les **hyperplans** non linéaires
- **C** : pénalité augmentant la distance des données prises en compte, au risque d'engendrer un surentraînement pour une valeur trop importante

## 2.1 Régression

Un hyperparamètre  $\epsilon$  permet de fait varier l'épaisseur de la marge, pour y inclure le plus de données possible. Les éléments exclus sont identifiés en rose.

### 2.1.1 Régression linéaire

Régression la plus simple : une approximation affine est suffisante.

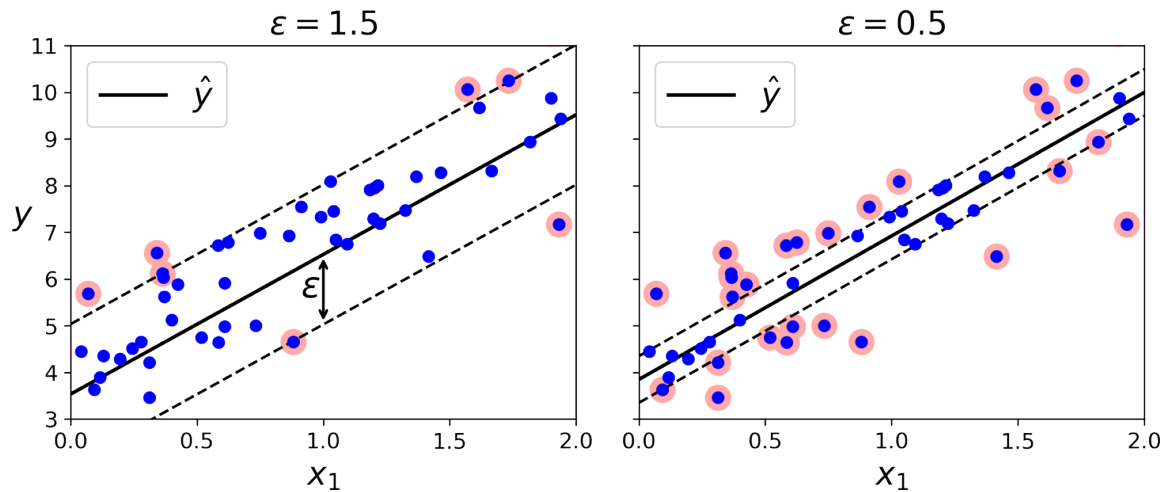


Fig. 1 : Régression linéaire, variation d' $\epsilon$  [5]

### 2.1.2 Régression non linéaire

Régression nécessitant l'utilisation d'une fonction noyau.  
Une plus grande valeur de  $C$  intègre des données plus éloignées.

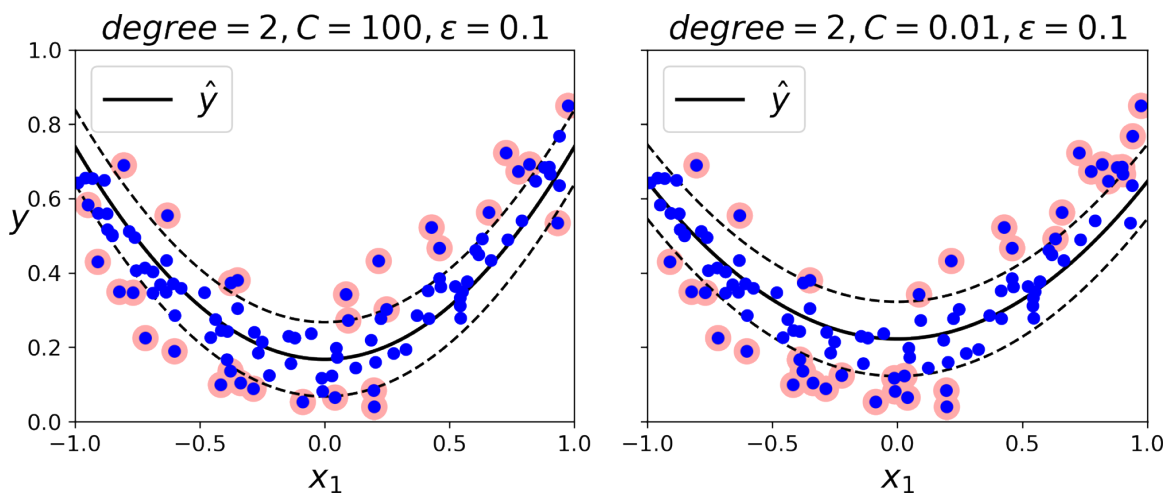


Fig. 2 : Régression polynomiale de degré 2, variation de  $C$  [6]

## 2.2 Classification

Il s'agit du type de résolution le plus fréquemment utilisé.

### 2.2.1 Classification linéaire

Cette section se penche sur la classification de 2 espèces d'iris, en fonction des longueurs et largeurs de leurs pétales.

#### Séparation à Vaste Marge

La figure de gauche montre que dans l'absolu, un grand nombre de droites peut séparer correctement les 2 ensembles à classifier. La figure de droite montre cependant qu'en utilisant les éléments les plus proches, appelés dans ce cas **support vectors**, il est alors possible de définir une marge de séparation la plus large qui soit, afin de déterminer la droite médiane de séparation la plus efficace.

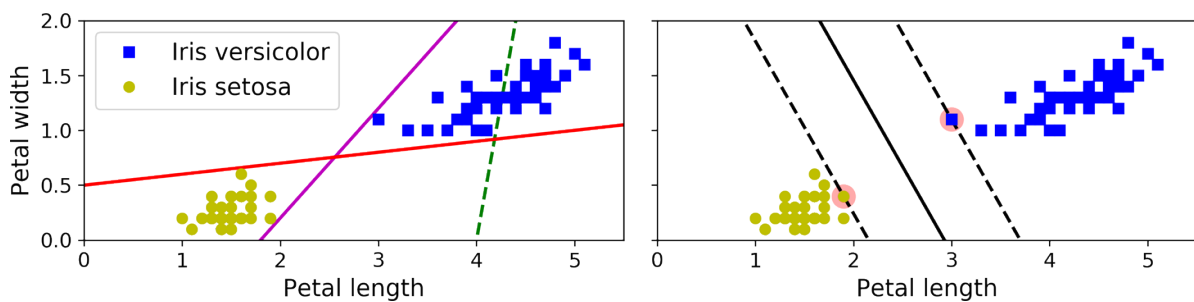


Fig. 3 : Séparation à Vaste Marge [7]

Un changement d'échelle préalable aide à la séparation des données, et peut mener à une meilleure efficacité du modèle pour la classification. [1]

La figure de droite montre l'inclusion d'un **support vector** supplémentaire.

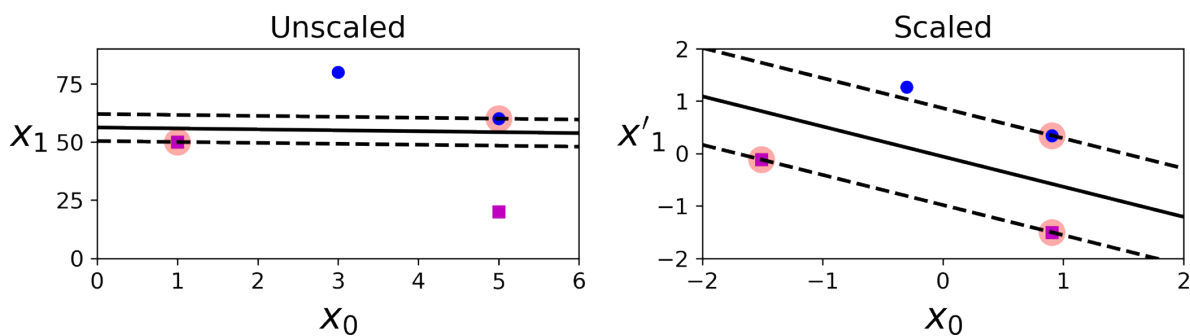


Fig. 4 : Changements d'échelles des dimensions [7]

## Séparation à marge souple

L'approche de vaste marge peut être perturbée par 2 problématiques distinctes.

La figure de droite montre par exemple des **support vectors** tellement proches, que la pertinence du modèle s'en trouve forcément impactée, réduisant ainsi la fiabilité de la séparation.

La figure de gauche montre quant à elle une anomalie (outlier), rendant de fait toute séparation linéaire impossible.

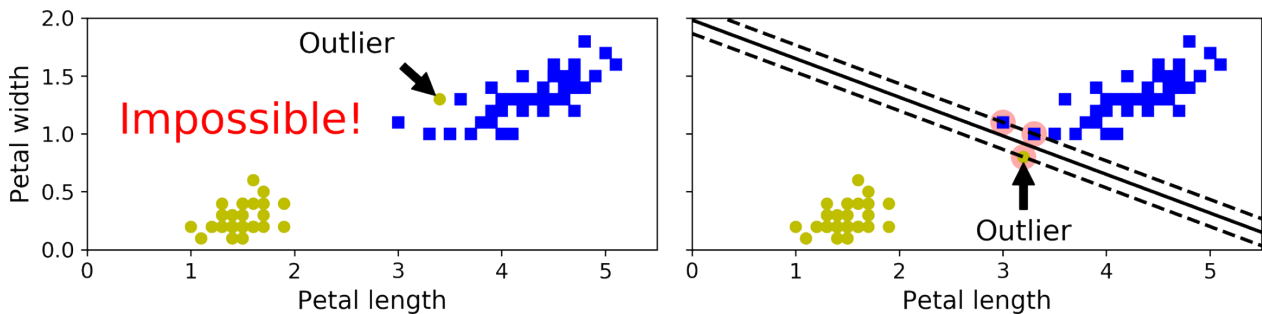


Fig. 5 : Sensibilité de vaste marge aux anomalies [8]

Il faut donc utiliser un modèle plus flexible pour pouvoir éviter ce type de problèmes. Le but étant de trouver le meilleur compromis entre avoir la marge la plus large, et y retrouver le moins possible d'éléments intrus, appelés violations de marge.

Il est ainsi possible d'utiliser l'hyperparamètre **C** pour faire varier la distance de prise en compte des éléments proches de la marge.

Plus la valeur de **C** augmente, plus le nombre de violations de marge diminue, mais plus le modèle se spécialise.

Une attention particulière doit donc être portée à la réduction de marge, sans quoi le modèle perdrait en capacité de généralisation et donc en précision.

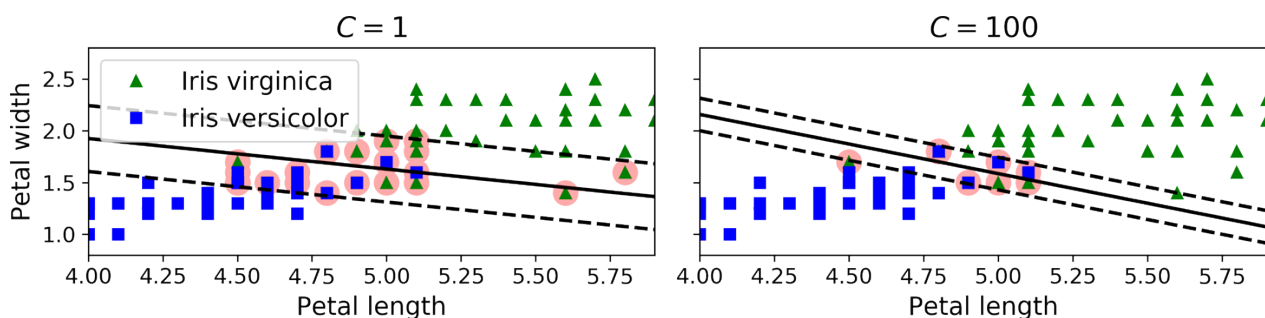


Fig. 6 : Plus ou moins de violations de marge, variation de C [8]

## 2.2.2 Classification non linéaire

Toutes les données ne sont pas forcément séparables de façon linéaire.

Pour y adapter un modèle, il est donc nécessaire de passer par la création de nouvelles variables résultant de la transformation des données de départ.

Sur la ligne de la figure de gauche, des éléments verts sont entourés d'éléments bleus, inséparables linéairement tels quels.

La figure de droite introduit alors une variable  $X_2$ , élévation au carré de la variable d'origine  $X_1$ , permettant ainsi une séparation linéaire des données.

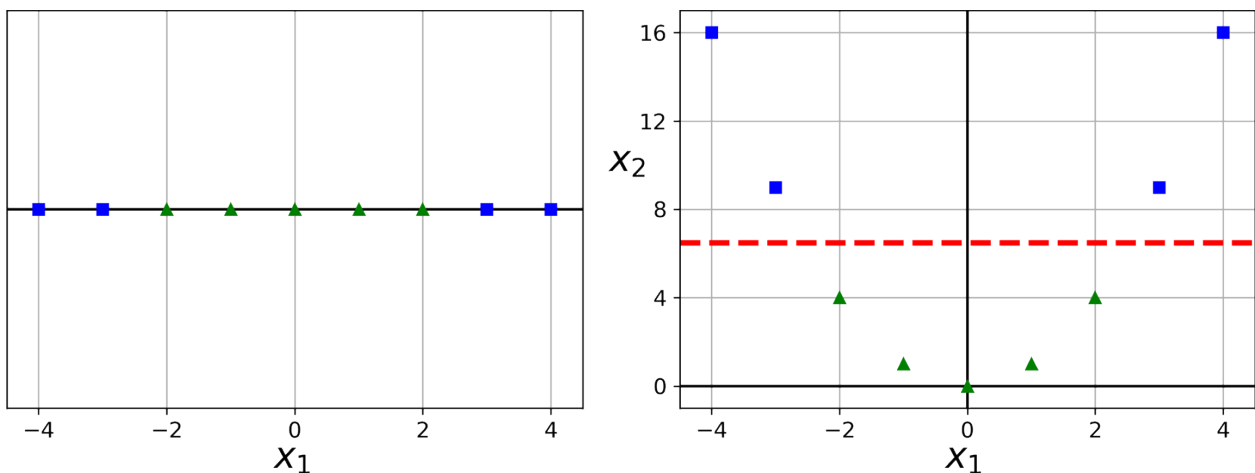


Fig. 7 : Séparation linéaire par ajout de variable [9]

L'ajout de variables polynomiales autorise donc des séparations sous forme de courbes, et non plus seulement de droites.

La figure suivante montre un exemple de séparation curviligne régulière, correspondant à la répartition des 2 catégories présentes.

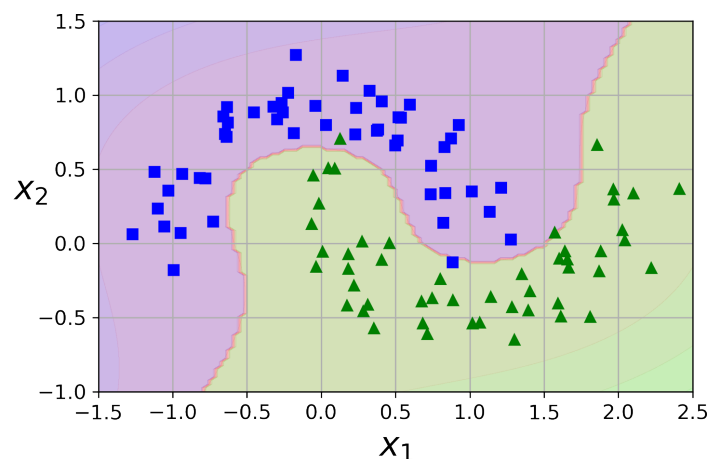


Fig. 8 : Classification utilisant des variables polynomiales [10]

## Noyau polynomial

Combiner des polynômes de degrés faibles est simple, pratique et relativement rapide à calculer.

Mais plus les [datasets](#) seront complexes, plus il faudra composer avec des polynômes de plus haut degré, générant un grand nombre de variables, et rendant ainsi le modèle trop lent.

Un outil mathématique appelé [kernel trick](#) permet de contourner ce problème. Se basant sur le théorème de Mercer, il permet de remplacer dans un espace de grande dimension, un produit scalaire par une [fonction noyau](#) facile à calculer.

Ce [kernel trick](#) simplifie donc les calculs ultérieurs d'affinage. La technique [grid search](#), limitant les hyperparamètres à des sous-ensembles de valeurs prédéfinies respectant un certain pas, permet ensuite un bon compromis de temps de calcul pour trouver les valeurs de réglages les plus appropriées.

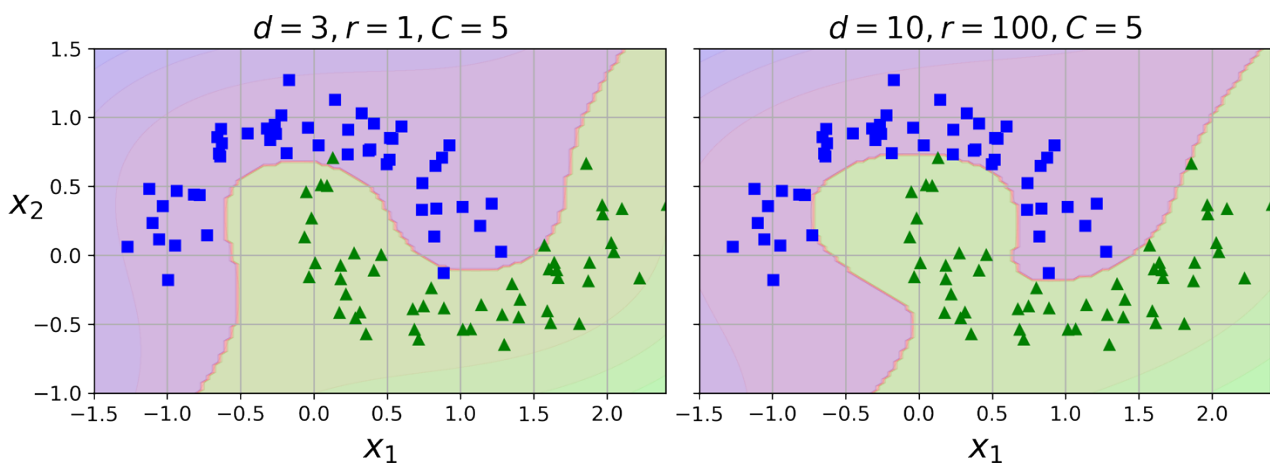


Fig. 9 : [Fonction noyau](#) polynomiale [11]

## Similarité

Une autre façon de gérer des données non linéairement séparables est d'utiliser une fonction de similarité, comme la [RBF](#) gaussienne.

Le principe est de transformer les éléments en fonction de leur similitude avec des points de repères déterminés dans l'ensemble d'origine. Utiliser chacun des éléments comme points de repère augmente la probabilité de séparabilité linéaire, mais également le nombre de variables et donc le temps de calcul.

La figure de gauche montre les 2 points de repère choisis en rouge, ainsi que leur courbe gaussienne associée.

La figure de droite montre tous les points transformés avec la [RBF](#) gaussienne, devenant également linéairement séparables.

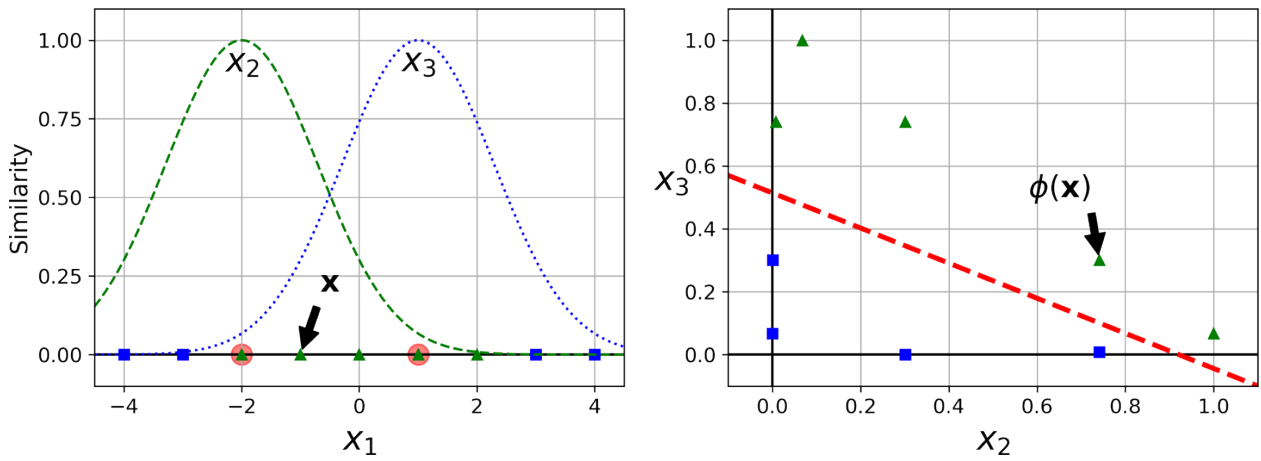


Fig. 10 : Variables de similarité utilisant la RBF gaussienne [12]

### Noyau gaussien RBF

Le **kernel trick** marche également dans un tel contexte de similarité.

Faire varier **C** (colonnes) change la distance de prise en compte.

Faire varier  **$\gamma$**  (rangées) modifie l'épaisseur de la « cloche » gaussienne.

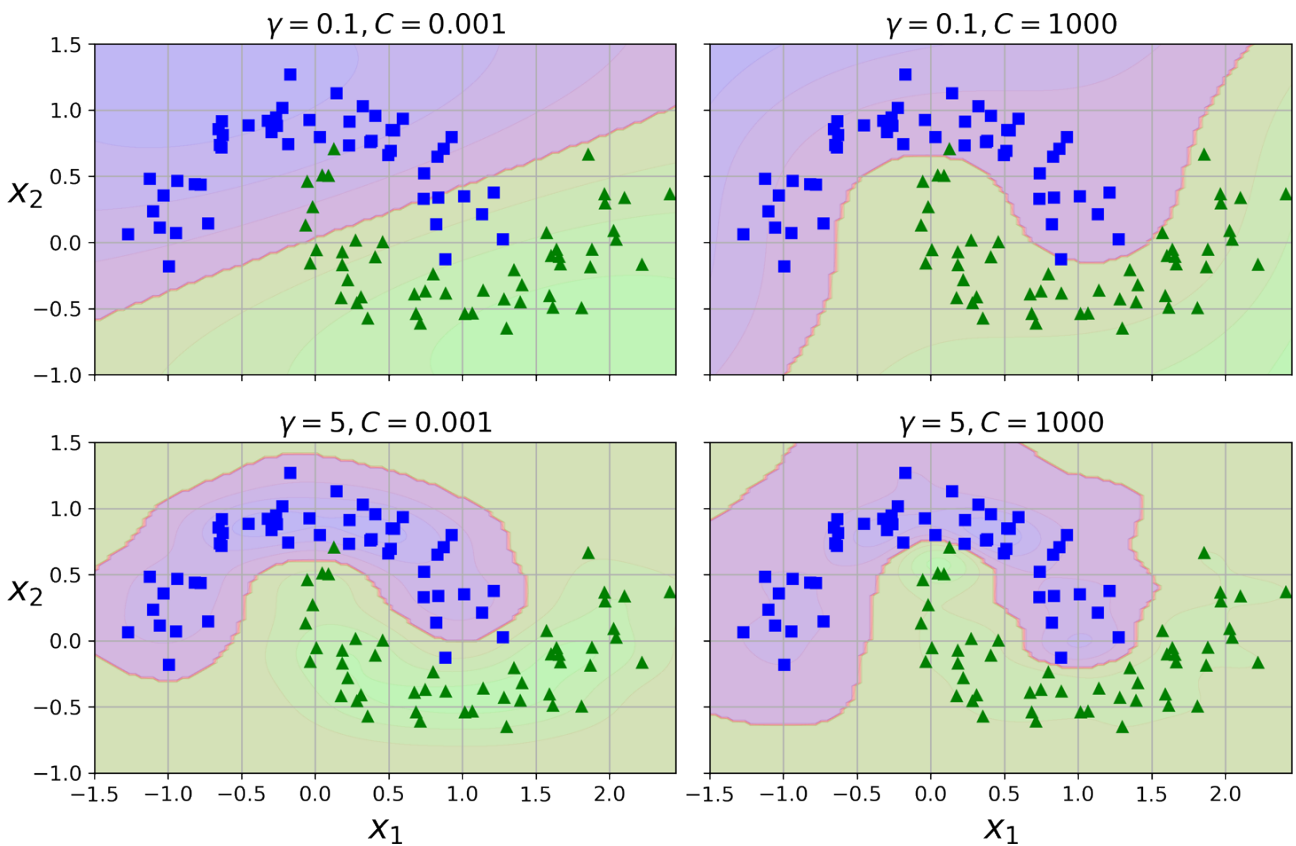


Fig. 11 : Fonction noyau gaussienne (RBF) [13]

## Autres noyaux

En complément aux plus courants vus précédemment, un certain nombre d'autres noyaux existe :

- [RBF](#) de Laplace
- tangente hyperbolique
- [sigmoïde](#)
- fonction de Bessel de première espèce
- [RBF ANalysis Of VAriance \(ANOVA\)](#)
- sillon linéaire à 1 dimension
- chaîne (utilisé pour les documents texte ou séquences d'ADN)

## Autres outils

Des algorithmes récents se sont montrés plus efficaces avec de grands [datasets](#) :

- descente par sous-gradient, avec des techniques de décomposition
- descente par coordonnée, en effectuant des minimisations itératives

### 2.2.3 Classification multi-classes

Quand au moins 3 catégories sont à classifier, et donc pour étendre les modèles à plus de 2 catégories, deux approches sont intéressantes [4] :

- [SVM](#) à arbre de décision
- [SVM](#) par paires

### 2.2.4 Optimisation

Normaliser les données avant de les classifier peut donner de bien meilleurs résultats en fonction du type de problèmes considéré.

Dans le cadre de la proposition d'un nouvel algorithme pour classifier des tumeurs cérébrales à partir d'[Imagerie par Résonance Magnétique \(IRM\)](#) [17], sont recensées (entre 1990 et 2018) de nombreuses combinaisons de pré-traitements et de classifications. [18]

Parmi ces traitements : filtre gaussien, réduction de bruit, suppression d'artefacts, égalisation d'histogramme, filtre médian.

### 3 Exemples

Les 2 cas suivants, linéaire [14] et non linéaire [15], illustrent bien le propos.

#### 3.1 Linéaire

Une classification est voulue pour pouvoir déterminer si un étudiant sera bon ou non en [machine learning](#), à partir des 2 notes qu'il a obtenu aux examens de Mathématiques et de Statistiques.

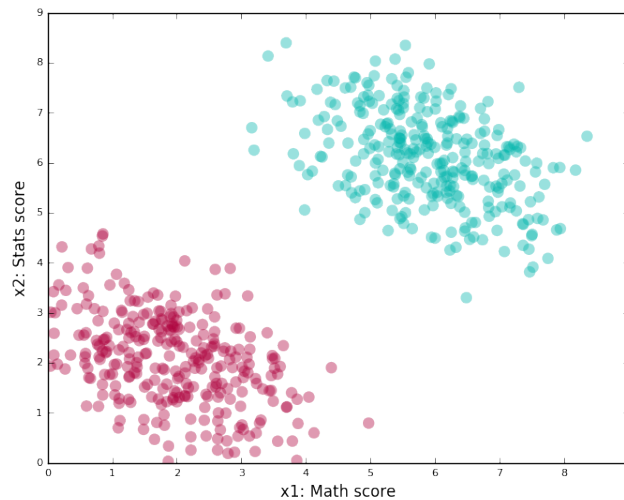


Fig. 12 : [Dataset](#) très bien distribué

Avec un tel [dataset](#), l'identification des [support vectors](#) est aisée, et une marge la plus large possible est facilement applicable.

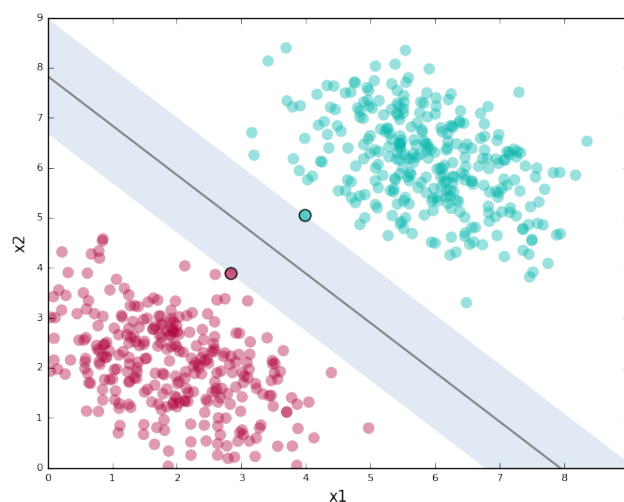


Fig. 13 : Séparation à Vaste Marge très nette

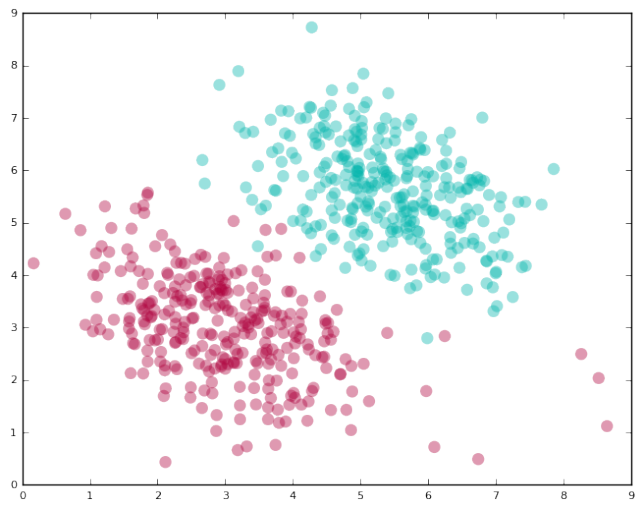


Fig. 14 : Anomalies dans le [dataset](#), violations de marge

Pour un [dataset](#) avec aberrations, faire varier une marge souple permet de trouver un compromis entre généralisation et spécialisation.

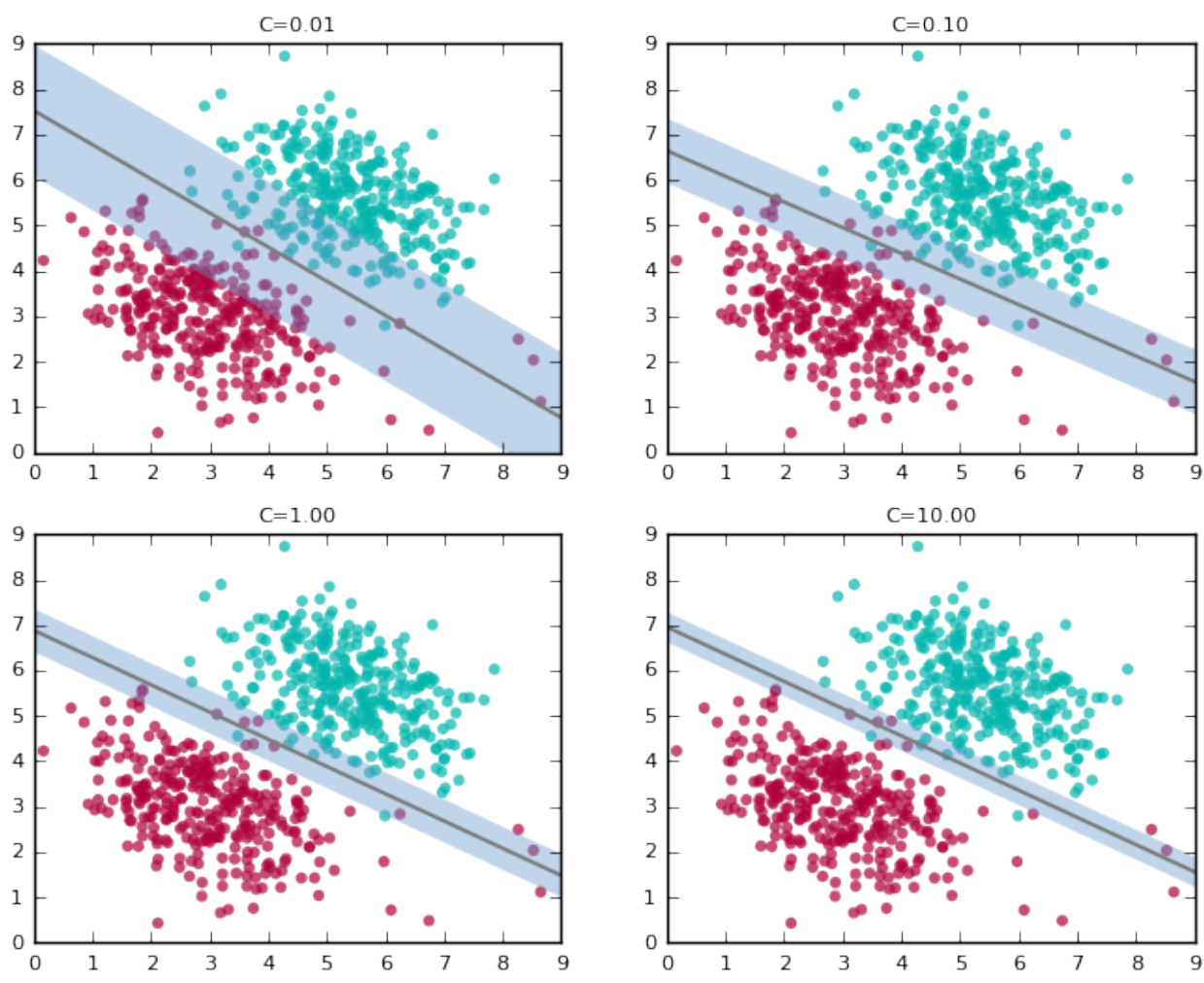


Fig. 15 : Différentes séparations à marge souple, variation de C

## 3.2 Non linéaire

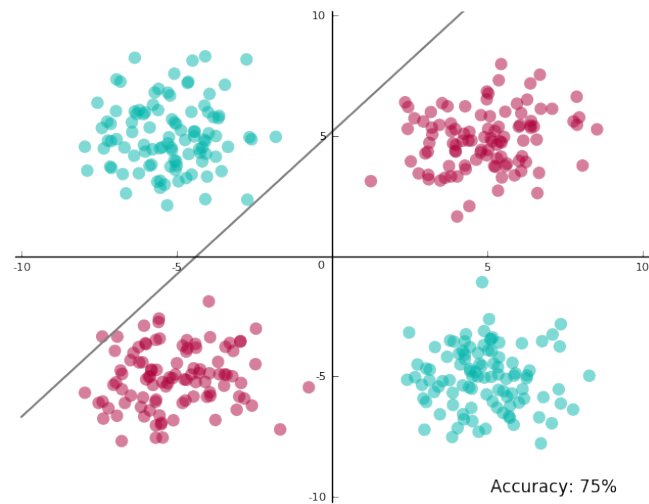


Fig. 16 : Dataset inséparable de façon linéaire

Création d'un nouvel espace dimensionnel avec des transformations non linéaires des variables d'origine.

$$X_1 = x_1^2 ; X_2 = x_2^2 ; X_3 = \sqrt{2} \times x_1 \times x_2$$

Ce nouvel espace permet de trouver une marge et un [hyperplan](#) séparateur.

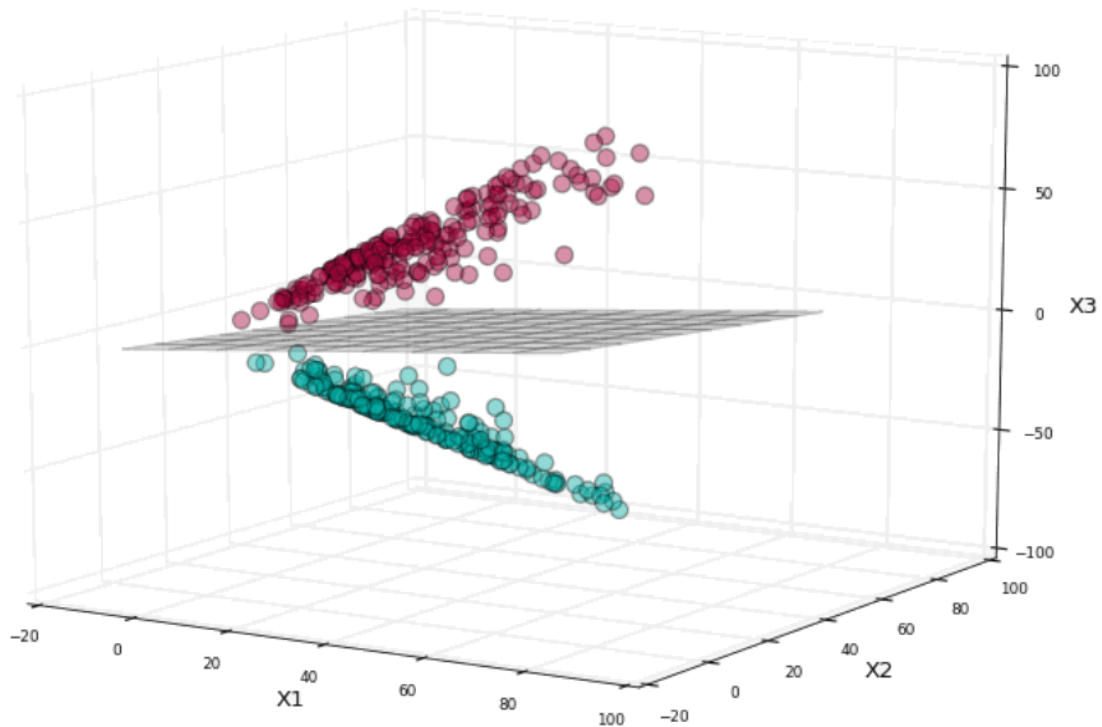


Fig. 17 : Hyperplan séparateur linéaire dans le nouvel espace dimensionnel

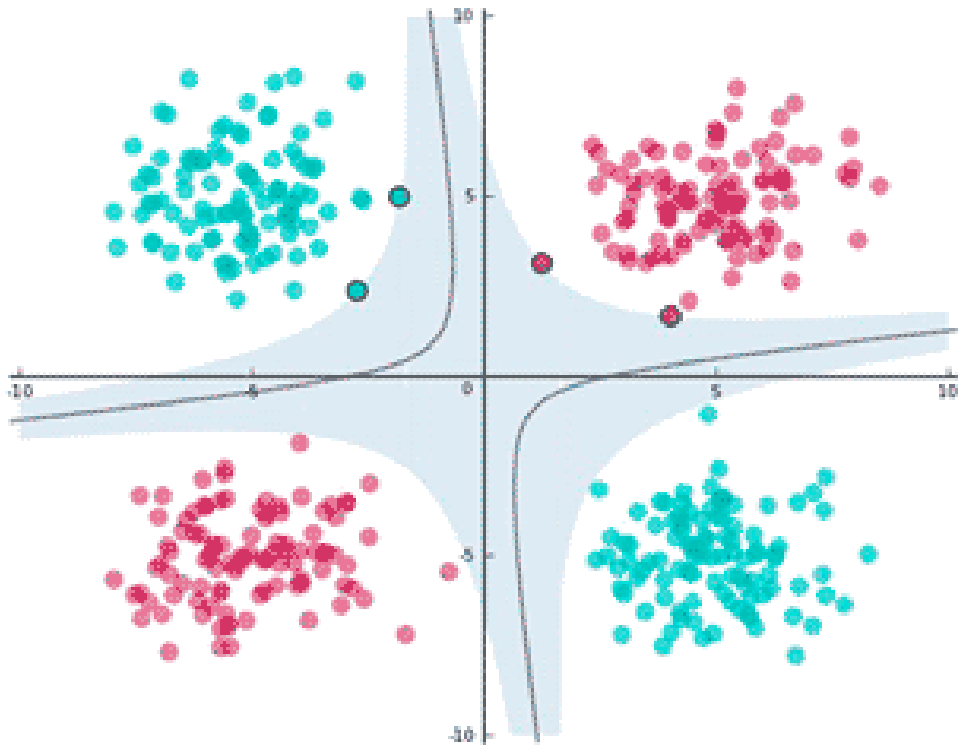


Fig. 18 : Projection de la marge de séparation dans l'espace d'origine

Le retour à l'espace de départ permet ainsi marge et séparation non linéaires.

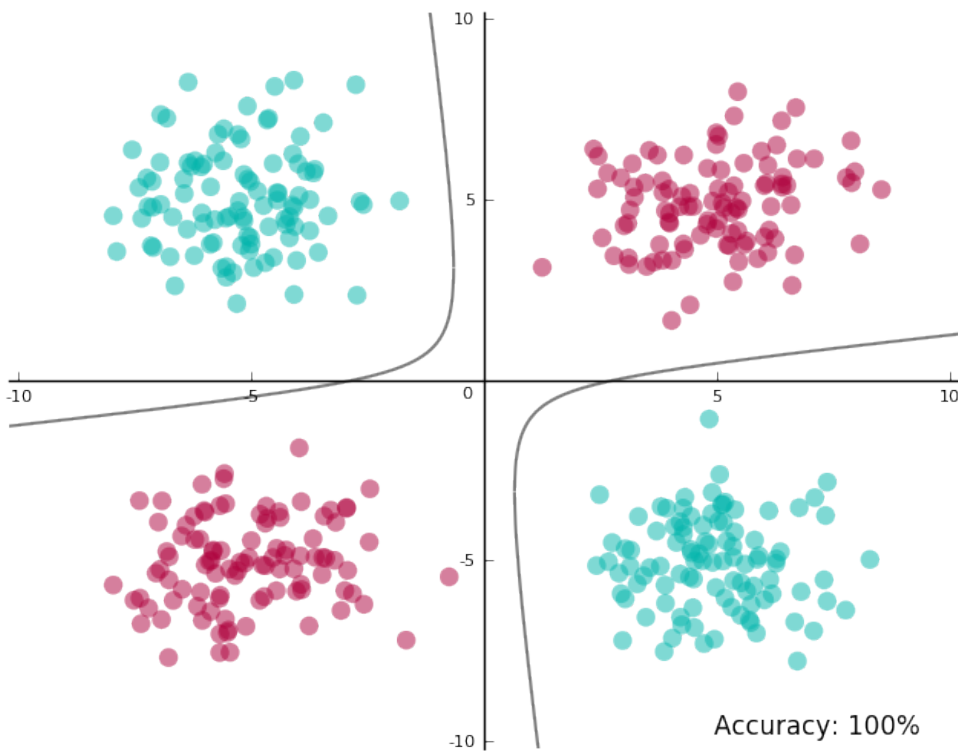


Fig. 19 : Fonction de décision finale non linéaire dans l'espace d'origine

## 4 Critiques

### 4.1 Avantages

- le modèle [SVM](#) est considéré comme plutôt stable, car de petits changements dans les données impactent peu les [hyperplans](#)
- ce même modèle peut être utilisé à la fois pour résoudre des problèmes de régression et de classification
- est souvent une meilleure alternative aux réseaux de neurones, surtout dans le cas de [datasets](#) de taille réduite
- très précis avec une marge de séparation nette
- peut capturer des relations complexes dans des [datasets](#), même sans connaissance préalable de ces dernières
- fonctionne avec un grand nombre d'espaces dimensionnels
- reste pertinent même avec plus de dimensions que d'éléments
- capacité de régularisation pour rester générique, afin d'éviter les risques de surapprentissage
- économise la mémoire nécessaire aux fonctions de décision, en ne leur faisant traiter que des [support vectors](#)
- permet de gérer efficacement des séparations non linéaires, par l'intermédiaire de différentes [fonctions noyau](#)
- de nombreuses [fonctions noyau](#) existent pour les fonctions de décision
- il est possible de combiner plusieurs [fonctions noyau](#), pour pouvoir travailler avec des [hyperplans](#) plus complexes
- pendant longtemps considéré inadapté aux très grands [datasets](#), de nouveaux algorithmes ont été mis à contribution depuis [2]

## 4.2 Inconvénients

- le choix d'une **fonction noyau** appropriée n'est pas évident, et peut facilement mener à un trop grand nombre de **support vectors**
- la mémoire nécessaire augmente avec le nombre de **support vectors**, car ces derniers doivent y être intégralement stockés
- difficile d'identifier les bonnes valeurs de paramètres, sans passer par des techniques comme **grid search**
- le temps d'entraînement augmente avec le nombre d'éléments
- les calculs de probabilité de justesse sont très coûteux, nécessitant une validation croisée en plusieurs étapes
- les modèles sont difficilement interprétables par des humains, contrairement par exemple aux arbres de décisions
- effet boîte noire en cas de compréhension insuffisante des différents outils mathématiques sous-jacents

## 4.3 Limitations

- fonctionne mal quand des classes se recouvrent, car plusieurs paramètres varient, contrairement à d'autres méthodes
- une normalisation préalable des données est nécessaire, pour que les fonctions objectifs soient plus pertinentes
- les problèmes multi-classes sont encore un champ de recherche à part entière

## 5 Conclusion

Les [SVM](#) sont incontournables dans le domaine du [machine learning](#), tant par leur polyvalence que leur flexibilité. Le nombre de pré-traitements, techniques, algorithmes et méthodes mathématiques impliqués en font un outil redoutablement efficace pour la régression et la classification de données.

De très nombreux langages de programmation permettent aujourd'hui de s'interfacer avec des bibliothèques implémentant déjà nombre de ces concepts (par exemple [libSVM](#) [3] ou encore [SVM-Light](#) [16]).

Même si certains éléments clés comme le [kernel trick](#) et les [fonctions noyau](#) sont bien ancrés, ce domaine est toujours un champ de recherche en constante activité. Au fil du temps, de nouvelles techniques et optimisations ont vu le jour, et d'autres viendront encore en bousculer les modèles...

## Références

- [1] Minaxi Arora et Lekha Bhambhu. « Role of Scaling in Data Classification Using SVM ». In : IJARCSSE 4 (10 2014), p. 271-273.
- [2] Antoine Bordes. « New Algorithms for Large-Scale SVM ». Thèse de doct. Université Paris VI, 2010.
- [3] Chih-Chung Chang et Chih-Jen Lin. LIBSVM. 2019. url : <https://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [4] Paresh Deka. A primer on machine learning applications in civil engineering. CRC Press, 2020, p. 86-89.
- [5] Aurélien Géron. Hands-on Machine Learning. 2nd. O'Reilly, 2019, p. 163.
- [6] Aurélien Géron. Hands-on Machine Learning. 2nd. O'Reilly, 2019, p. 164.
- [7] Aurélien Géron. Hands-on Machine Learning. 2nd. O'Reilly, 2019, p. 154.
- [8] Aurélien Géron. Hands-on Machine Learning. 2nd. O'Reilly, 2019, p. 155.
- [9] Aurélien Géron. Hands-on Machine Learning. 2nd. O'Reilly, 2019, p. 157.
- [10] Aurélien Géron. Hands-on Machine Learning. 2nd. O'Reilly, 2019, p. 158.
- [11] Aurélien Géron. Hands-on Machine Learning. 2nd. O'Reilly, 2019, p. 159.
- [12] Aurélien Géron. Hands-on Machine Learning. 2nd. O'Reilly, 2019, p. 160.
- [13] Aurélien Géron. Hands-on Machine Learning. 2nd. O'Reilly, 2019, p. 161.
- [14] Abhishek Ghose. Learning SVMs From Examples. 2017. url : <https://www.kdnuggets.com/2017/08/support-vector-machines-learning-svms-examples.html>.
- [15] Abhishek Ghose. Learning SVMs From Examples. 2017. url : <https://www.kdnuggets.com/2017/08/support-vector-machines-learning-svms-examples.html/2>.
- [16] Thorsten Joachims. SVMlight. 2008. url : [http://www.cs.cornell.edu/people/tj/svm\\_light](http://www.cs.cornell.edu/people/tj/svm_light).
- [17] Anand Kulkarni et Suresh Satapathy. Optimization in Machine Learning and Applications. Springer, 2020, p. 51-68.
- [18] Anand Kulkarni et Suresh Satapathy. Optimization in Machine Learning and Applications. Springer, 2020, p. 54.

# Figures

1	Régression linéaire, variation d' $\epsilon$ [5]	7
2	Régression polynomiale de degré 2, variation de C [6]	7
3	Séparation à Vaste Marge [7]	8
4	Changements d'échelles des dimensions [7]	8
5	Sensibilité de vaste marge aux anomalies [8]	9
6	Plus ou moins de violations de marge, variation de C [8]	9
7	Séparation linéaire par ajout de variable [9]	10
8	Classification utilisant des variables polynomiales [10]	10
9	Fonction noyau polynomiale [11]	11
10	Variation de similarité utilisant la RBF gaussienne [12]	12
11	Fonction noyau gaussienne (RBF) [13]	12
12	Dataset très bien distribué	14
13	Séparation à Vaste Marge très nette	14
14	Anomalies dans le dataset, violations de marge	15
15	Différentes séparations à marge souple, variation de C	15
16	Dataset inséparable de façon linéaire	16
17	Hyperplan séparateur linéaire dans le nouvel espace dimensionnel	16
18	Projection de la marge de séparation dans l'espace d'origine	17
19	Fonction de décision finale non linéaire dans l'espace d'origine	17



## **SVM**

### **Machine à Vecteurs de Support**

**Mémoire probatoire présenté en vue d'obtenir  
UE « Information et communication pour ingénieur »  
Spécialité :  
Informatique, Réseaux, Systèmes et Multimédia**

**Bordeaux, 2020**

---

#### **RÉSUMÉ**

La croissance constante du [big data](#) nécessite de nouvelles méthodes d'analyse. Le [machine learning](#) propose une famille de méthodologies, parmi laquelle les [SVM](#) permettent à la fois régression et classification de ces données.

La polyvalence et la flexibilité des [SVM](#) les ont donc rendus incontournables. D'autant que de nombreux langages et bibliothèques permettent de les utiliser. Au fil du temps, ces outils ont toujours été améliorés, et continuent de l'être...

**Mots clés : svm, noyau, linéaire, régression, classification, apprentissage, machine, données.**

---

#### **SUMMARY**

The constant growth of big data requires new methods of analysis. [Machine learning](#) provides a whole family of dedicated methodologies, including [SVM](#) that allow both regression and classification of [datasets](#).

The flexibility and versatility of [SVM](#) therefore made them indispensable. Especially since now many languages and libraries enable their usage. Over time, these tools have always been and continue to be improved...

**Key words : svm, kernel, linear, regression, classification, machine, learning, datasets.**