UNIVERSITÉ DE MONTPELLIER

Tp apprentissage statistique

Quentin Festor

September 2024

Table des matières

1 Introduction aux Svm

Les machines à vecteurs de support (SVM, pour Support Vector Machines) sont une famille d'algorithmes d'apprentissage supervisé largement utilisés pour résoudre des problèmes de classification et de régression. Leur principe repose sur la recherche de la meilleure séparation possible entre différentes classes de données à l'aide d'hyperplans.

Comme énoncé précédemment, l'objectif principal d'un SVM est de trouver un hyperplan (une frontière) qui sépare les classes de manière optimale, en maximisant la marge entre les points les plus proches de chaque classe, appelés vecteurs de support. Ce type de modèle est particulièrement apprécié pour sa capacité à gérer des données complexes, voire non linéaires, grâce à l'utilisation de la méthode du noyau.

Cas linéaire : Si les données sont linéairement séparables, l'algorithme trouve un hyperplan qui sépare les classes en maximisant la distance entre les vecteurs de support des deux classes.

Cas non linéaire : Si les données ne peuvent pas être séparées par une simple ligne (ou un plan dans des dimensions plus élevées), les SVM utilisent des fonctions noyau (kernel functions) pour transformer les données dans un espace de plus haute dimension où une séparation linéaire devient possible. On a alors plusieurs possibilités, nous utiliserons principalement les noyaux linéaire, polynomiale et gaussien.

Dans ce tp, l'objectif sera d'explorer l'application des SVM en comparant les performances des modèles avec les différents noyaux et en faisant varier les hyperparamètres afin de comprendre les impacts de ces manipulations.

Pour ce compte rendu, il a été décidé de laisser le code en annèxe afin de ne pas gacher la rédaction. On n'affichera dans la partie principale seulement les résultats commentés ainsi que les graphiques, sauf si l'apparition du code semble pertinante pour la rédaction.

2 Mise en oeuvre

2.1 Kernel linéaire

Nous reprenons le code donné en annexe concernant l'utilisation des SVM et l'appliquons à nos données Iris. Tout d'abord, nous classifions la classe 1 contre la classe 2 en utilisant uniquement les deux premières variables caractéristiques et un noyau linéaire. Les données sont ensuite séparées de manière équitable en deux échantillons : un pour l'entraînement et un pour le test, afin d'évaluer la qualité de notre modèle.

Nous évaluons ensuite la performance du modèle sur les données d'entraînement, puis sur les données de test, et obtenons les scores suivants :

Kernel Type	**Training Score**	**Test Score**
Linear Kernel	0.68	0.64

Table 1 – Score pour le kernel linéaire SVM.

L'analyse de ces résultats montre que le modèle SVM avec un noyau linéaire a une précision de 0.68 sur les données d'entraînement, ce qui indique une adéquation initiale correcte du modèle. Cependant, une légère baisse de performance est observée sur les données de test, avec une précision de 0.64, ce qui peut indiquer que le modèle a un léger problème de généralisation. Cela suggère que le modèle ne capture peut-être pas parfaitement la complexité des données, ou qu'il y a peut-être un léger surapprentissage sur l'ensemble d'entraînement.

Ce résultat nous donne envie de de tester un autre noyau afin de vérifier si un noyau différent améliorerait la capacité de généralisation.

2.2 Kernel polynomial

Après avoir observé des résultats mitigés avec le noyau linéaire, nous décidons d'utiliser un noyau polynomial et de comparer ses performances. Nous utilisons évidemment la même séparation des données pour garantir une comparaison juste entre les deux approches.

Les paramètres optimaux trouvés avec la validation croisée sont :

Parameter	Value
C	0.001
Degree	2
Gamma	10.0
Kernel	Polynomial
Training Score	0.70
Test Score	0.52

TABLE 2 – Paramètres obtenus par cross-validation pour le kernel polynomial SVM.

Le modèle SVM avec un noyau polynomial affiche une précision de 0,70 sur l'ensemble d'entraînement et de 0,52 sur l'ensemble de test. Par rapport au noyau linéaire, nous constatons une légère amélioration de la précision sur les données d'entraînement (de 0,68 à 0,70), ce qui suggère que le noyau polynomial parvient à mieux capturer certaines interactions non linéaires présentes dans les données d'entraînement. En revanche, on observe une forte diminution de la précision sur les données de test (de 0,64 à 0,52), ce qui indique que le noyau polynomial est moins adapté à ce type de données et entraîne un surapprentissage. Dans ce cas précis, le noyau polynomial semble mal généraliser le modèle pour de nouvelles données.

Le noyau linéaire reste donc une option solide pour ce type de problème.

Les graphiques ci-dessous illustrent les deux classifications des données de l'ensemble Iris utilisées ci-dessus.

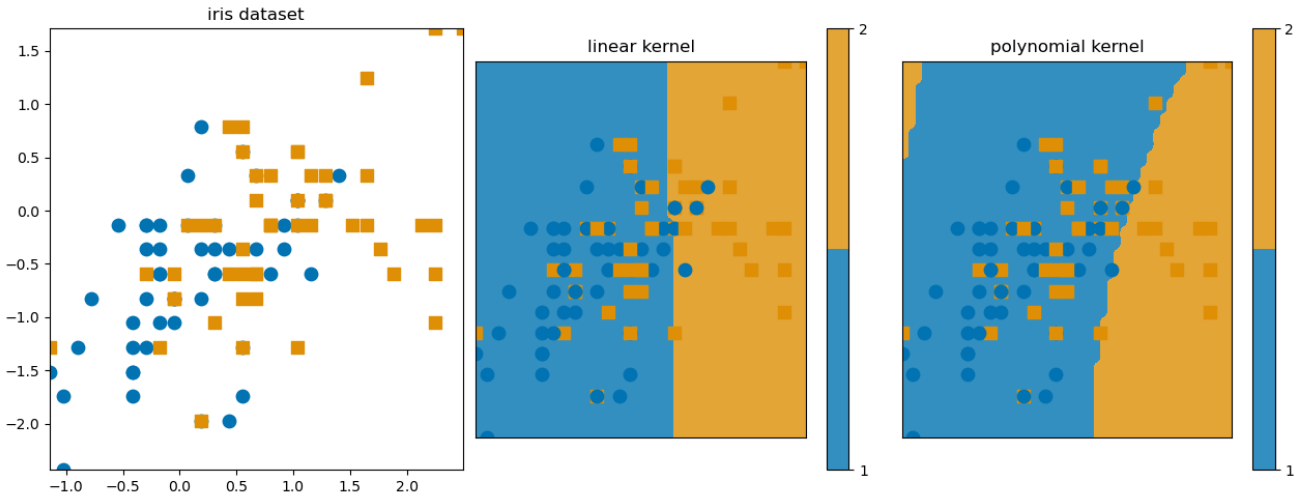

FIGURE 1 – Utilisation du modèle Svm sur nos données Iris avec un noyau linéaire, et un noyau polynomial

3 Classification de visages

Désormais on utilise une seconde base de données, comprenant des images de visages, et on souhaite classifier ces visages. D'abord on ne garde que les personnes possédant au moins 70 photos de leur visage, puis on redimensionne la taille des images car celles-ci prennent beaucoup de place. Le code a pour objectif de télécharger un sous-ensemble du dataset LFW, de sélectionner les images de deux personnes spécifiques (Tony Blair et Colin Powell) et de préparer ces données pour une tâche de classification binaire. On peut visualiser un échantillon des images ci-dessous :

FIGURE 2 – Echantillon de notre base de données pour Tony Blair et Colin Powell

3.1 Influence du paramètre de régularisation

Comme précédemment, on sépare nos données en deux échantillons de tailles distinctes pour avoir un échantillon d'entrainement et un échantillon de test. On veut montrer l'influence du paramètre de régularisation. Pour ce faire, on décide d'afficher le score d'apprentissage sur une échelle logarithmique entre 1e5 et 1e-5., et on obtient :

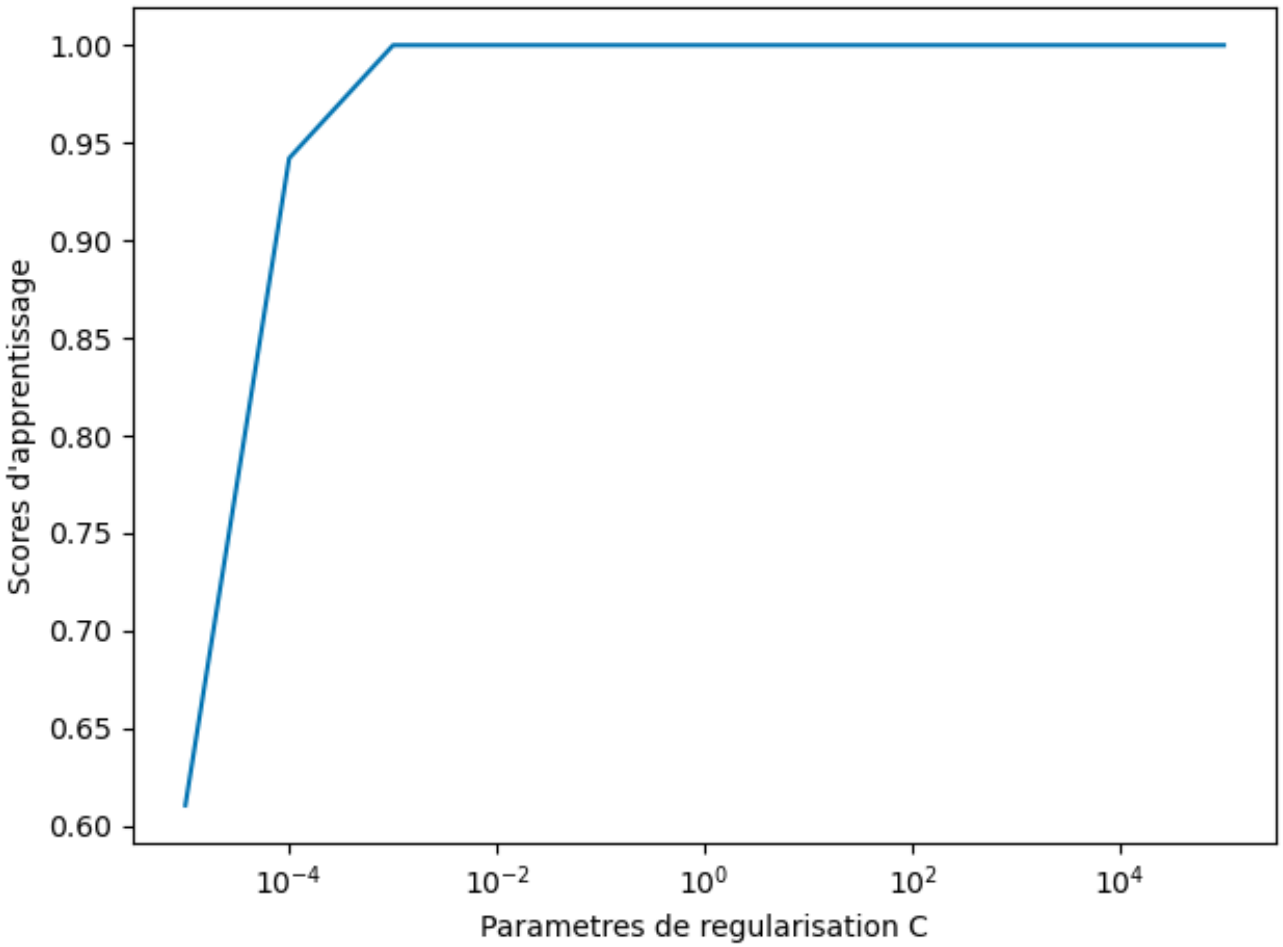

FIGURE 3 – Influence du paramètre de régularisation C sur le score d'apprentissage

Le meilleur paramètre C est celui qui donne le score de test le plus élevé. Dans ce cas, le meilleur C est 10^{-3} avec un score de 1. Pour des valeurs de C plus grandes, le score reste stable ce qui indique que les performances se stabilisent.

Maintenant que nous avons identifié le meilleur C, nous réentraînons le modèle avec cette valeur, et nous effectuons des prédictions finales pour l'ensemble de test. Nous comparons ensuite la précision obtenue au niveau de chance (prédiction aléatoire).

Résultat	**Valeur**
Précision du modèle (test)	0.93
Taux de chance (aléatoire)	0.62

TABLE 3 – Comparaison entre la précision du modèle SVM et le taux de chance.

Nous obtenons que la précision du modèle pour les données de test est de 0.93, alors que le taux de chance (niveau aléatoire) est de 0.62. Cela signifie que le modèle a une performance bien supérieure au hasard, ce qui indique qu'il a bien appris à distinguer les classes à partir des caractéristiques fournies.

L'échantillon d'images du premier document ci-dessous illustre les résultats de l'algorithme.

FIGURE 4 – Echantillon des prédictions du Svm sur nos données

La figure 5 montre les coefficients du modèle linéaire appris sous la forme d'une carte de chaleur. Cela permet de visualiser quelles parties des images sont les plus importantes pour le classifieur afin de prendre ses décisions. Ces zones correspondent à des caractéristiques importantes identifiées par le modèle dans les images.

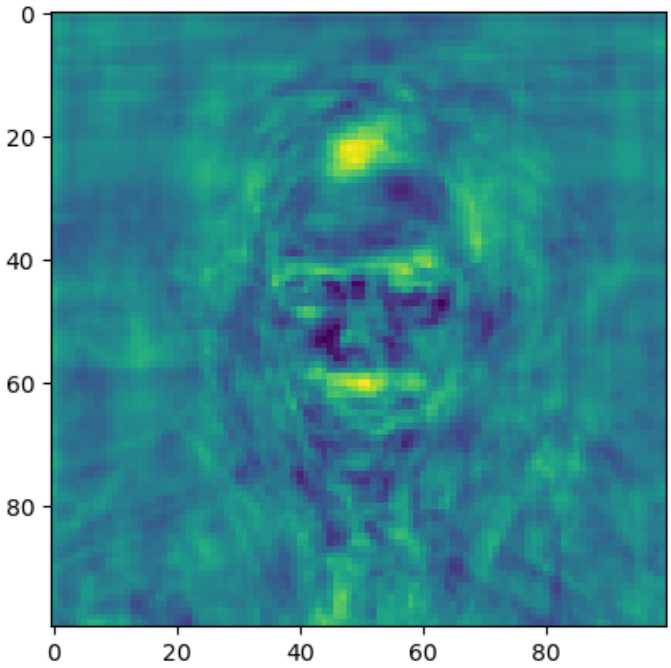

FIGURE 5 – Identification des zones importantes du visage pour la modélisation

Les zones importantes pour la prédiction semblent être l'implantation des cheveux au niveau du front, le nez, les yeux, ainsi que la bouche et la forme du crâne.

3.2 Ajout de nuisance sur les variables

Ajouter des variables de nuisance dans les données d'apprentissage est une méthode pour évaluer la robustesse et la capacité de généralisation d'un modèle de classification. Les variables de nuisance sont des caractéristiques ajoutées artificiellement qui n'ont aucune relation avec la variable cible. Il peut y avoir plusieurs objectifs à cet ajout. L'objectif principal est d'examiner l'impact de ces variables sur la performance du modèle et de vérifier si le modèle peut encore faire des prédictions précises malgré la présence de ces informations non pertinentes.

Condition	Précision (Entraînement)	Précision (Test)
Sans variables de nuisance	1.0	0.9
Avec variables de nuisance	1.0	0.56

Table 4 – Performances du modèle avec et sans variables de nuisance.

Sans ajout de variables de nuisance, le modèle atteint une précision parfaite de 1.0 sur l'ensemble d'entraînement, et une précision de 0.9 sur l'ensemble de test. Cela indique que le modèle est très performant et capable de bien généraliser sur des données non vues.

Lorsque des variables de nuisance sont ajoutées, la précision sur l'ensemble de test chute considérablement à environ 0.51, près du niveau de chance pour ce problème (0.62). Cependant, la précision sur l'ensemble d'entraînement reste à 1.0. Ce résultat montre que, bien que le modèle continue à bien performer sur les données d'entraînement, il est moins capable de généraliser sur de nouvelles données lorsqu'il est confronté à des informations non pertinentes.

3.3 Réduction des dimensions, PCA

Après avoir examiné l'impact des variables de nuisance sur la performance du modèle, il est pertinent de se pencher sur une autre technique importante de prétraitement des données : la réduction de dimensions. En utilisant l'Analyse en Composantes Principales (PCA), nous cherchons à réduire le nombre de caractéristiques tout en conservant autant d'information que possible. Cette méthode peut aider à améliorer la performance du modèle en éliminant le bruit et en simplifiant la représentation des données. Nous allons donc appliquer la PCA sur les données perturbées par le bruit et évaluer comment cette réduction de dimension influence la précision du modèle SVM. On essaie une fois avec 20 composantes, une fois avec 80, et une fois avec 120.

Composantes PCA	Score sur les données d'entrainements	Score dsur les données de test
5	0.6052	0.6157
10	0.6052	0.6368
15	0.6526	0.5894
20	0.6578	0.5894
25	0.6947	0.5842
80	0.7473	0.4894
120	0.8474	0.5263
200	0.9263	0.5210

Table 5 – Scores d'apprentissage après réduction de dimension par PCA avec un noyau linéaire

L'impact de la réduction de dimension par PCA sur les scores d'apprentissage montre une tendance globale. En réduisant le nombre de composantes, on observe une diminution progressive du score sur les données d'entraînement (ce qui reflète la capacité du modèle à bien capturer la variance des données), tout en voyant parfois une légère amélioration sur les données de test pour un certain intervalle de composantes. Toutefois, cette amélioration sur les données de test n'est pas linéaire. Par exemple :

Avec 5 à 10 composantes, le score de généralisation s'améliore légèrement, mais au-delà de 15 composantes, on observe une détérioration du score sur les données de test. Au fur et à mesure que le nombre de composantes augmente (notamment à partir de 25 composantes), le modèle tend à sur-apprendre (score parfait sur les données d'entraînement avec 120 composantes), mais il perd sa capacité de généralisation sur les données de test.

Ainsi, il apparaît qu'une réduction modérée du nombre de composantes permet d'améliorer la capacité de généralisation du modèle, probablement en limitant l'overfitting. Cependant, un nombre trop élevé de composantes réintroduit de la complexité, ce qui peut mener à une sur-adaptation aux données d'entraînement et à une baisse des performances sur les données de test.

La sélection optimale du nombre de composantes doit donc chercher à minimiser le nombre de composantes tout en conservant un score de généralisation décent. Cela permettra d'améliorer la robustesse du modèle sans le rendre trop spécifique aux données d'entraînement.

4 Conclusion

Pour résumer, l'objectif avec les SVM est de maximiser les performances de prédiction en choisissant un noyau optimal et en ajustant les hyperparamètres. Après avoir divisé les données en échantillons d'entraînement et de test, un score élevé sur l'ensemble d'entraînement indique un bon ajustement initial, mais seul un score élevé sur l'ensemble de test garantit que le modèle généralisera bien sur de nouvelles données. L'ajustement du paramètre de régularisation peut améliorer la précision. Nous avons également observé que l'ajout de variables de nuisance réduit les performances, mais ces dernières peuvent être restaurées en appliquant une réduction de dimension via l'analyse en composantes principales (PCA).