HAX907X Apprentissage statistique

TP:Support Vector Machines (SVM)

Marine GERMAIN
Coralie ROMANI DE VINCI

Table des matières

1 Introduction

Les *Support Vector Machines* (SVM) sont des méthodes de classification supervisée visant à séparer les différentes classes en maximisant la marge entre elles. Elles reposent sur la recherche d'hyperplans séparateurs dans un espace de grande dimension.

Dans ce TP, nous allons mettre en pratique ces techniques de classification sur des données réelles et simulées, en utilisant le package `scikit-learn`. Nous étudierons également l'influence des hyper-paramètres et des noyaux. Nous nous appuyons sur le dataset `iris` et une base de données de visages.

2 Base de données `iris`

Dans cette section, nous étudierons la base de données `iris` sur laquelle nous ferons une étude de classification sur les classes 1 et 2. Nous utiliserons d'abord le noyau linéaire puis le noyau polynomial et nous les comparerons. Nous avons séparé le dataset en deux parties:un ensemble d'entraînement (75% des données) et un ensemble de test(25% des données).

2.1 Classification avec noyau linéaire

La classification des deux premières variables avec un noyau linéaire a pour objectif de trouver un hyperplan qui sépare au mieux les deux classes en maximisant la marge entre les points les plus proches de chaque classe. Les scores obtenus pour cette méthode sont les suivants:

```
{'C': np.float64(0.013049019780144023), 'kernel': 'linear'}
Generalization score for linear kernel: 0.6933333333333334, 0.64
```

Figure 1: Score obtenu pour le noyau linéaire

Pour les données d'entraînement, le modèle a classifié correctement 69,3% des données, contre 64% pour les données de test.

2.2 Classification avec noyau polynomial

La classification des deux premières variables avec un noyau polynomial a pour objectif de trouver une frontière de décision non linéaire capable de séparer au mieux les deux classes. Nous souhaitons ici voir si l'emploi d'un noyau polynomial nous permet d'obtenir un meilleur classifieur que celui obtenu précédemment.
Les scores obtenus pour cette méthode sont les suivants :

```
{'C': np.float64(1.0), 'degree': np.int64(1), 'gamma': np.float64(10.0), 'kernel': 'poly'}
Generalization score for polynomial kernel: 0.68, 0.68
```

Figure 2: Score obtenu pour le noyau polynomial

Pour les données d'entraînement et de test, le modèle a classifié correctement 68% des données.

Comparaison des deux méthodes :

- Nous disposons d'abord des scores obtenus en figure1 et figure2. On remarque que le score pour le noyau linéaire est très légèrement plus élevé que celui du noyau polynomial pour les données d'entraînements 0.693 contre 0.68. En revanche il est légèrement plus faible pour les données de test 0.64 contre 0.68. A noter que ces différences sont faibles et que ces deux méthodes semblent donc similaires.

- Ensuite, nous avons tracé graphiquement ces résultats à l'aide des frontières :

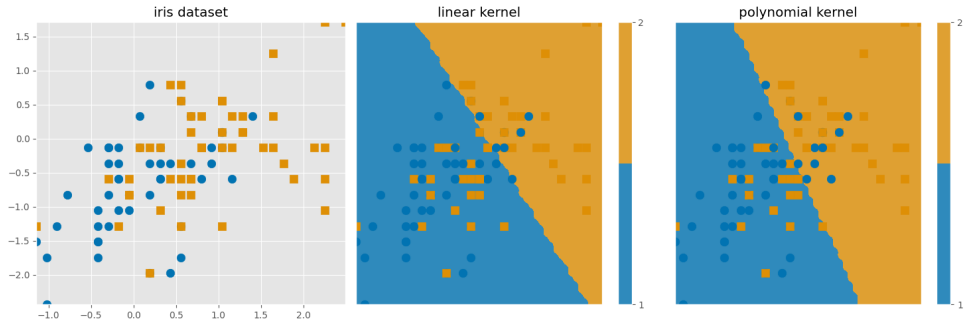
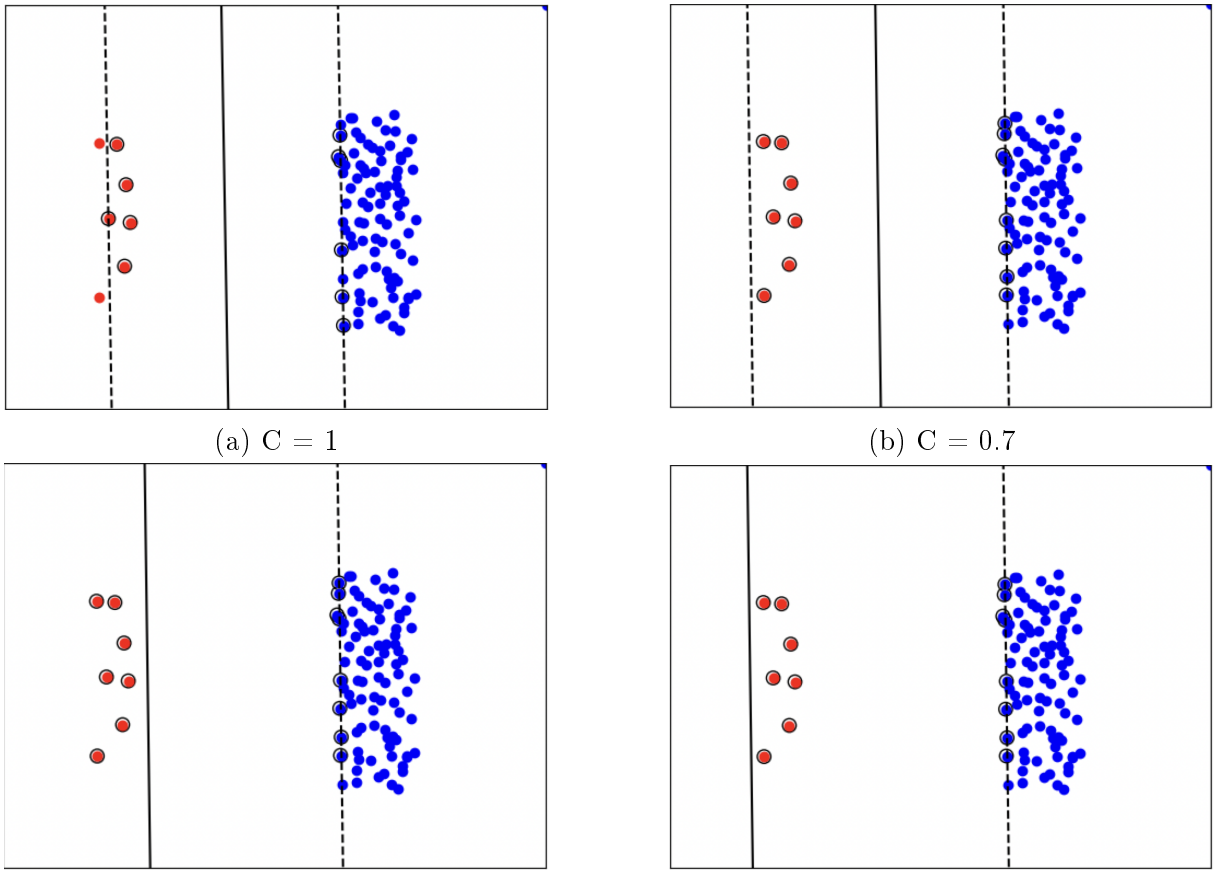


Figure 3: Visualisation de la frontière séparant les classes 1 et 2 pour les noyaux linéaire et polynomial

La figure3 permet de visualiser les deux classifieurs sur l'ensemble des données du dataset `iris`.
On observe que la frontière du noyau polynomial semble plutôt linéaire. La différence remarquable est que pour le noyau linéaire on observe qu'il y a moins de données "bleues" mal classifiées que pour le noyau polynomial. On peut faire l'observation inverse concernant les données "oranges". A REFORMULER + CONCLURE

3 SVM GUI

Nous utilisons dans cette section le script `svm_gui.py`.



(a) C = 1 (b) C = 0.7

(c) C = 0.4 (d) C = 0.3

Figure 4: Visualisation de la région de décision selon différents paramètres de régularisation C

CONCLURE

4 Classification des visages

Dans cette section, nous exploitons une base d'images extraites de « Labeled Faces in the Wild ». L'objectif est de classifier, à l'aide d'un SVM à noyau linéaire, deux types d'images : des portraits de Tony Blair (figure 5) et de Colin Powell.

Figure 5: Base de données de 12 portraits de Tony Blair

4.1 Influence du paramètre de régularisation

Nous souhaitons observer l'influence du paramètre de régularisation C sur la qualité du classifieur. Pour cela, nous estimons le score d'apprentissage en fonction de ce paramètre en figure 6.

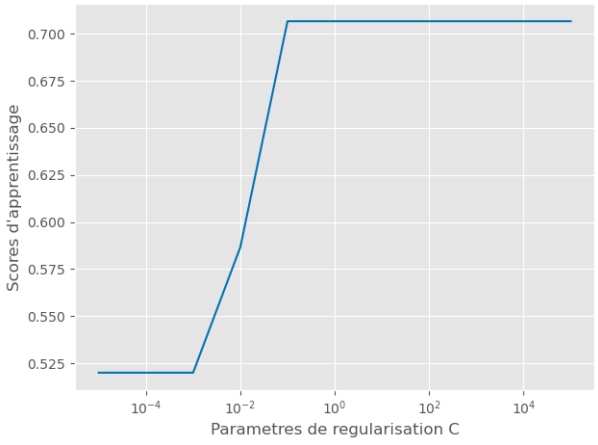


Figure 6: Score d'apprentissage en fonction du paramètres de régularisation C pour des classifieurs SVM à noyau linéaire

En faisant varier les valeurs de C entre 10^{-5} et 10^{5}, on remarque qu'on atteint un score d'apprentissage maximal d'environ 0.725 lorsque $C = 10^{-1}$. Au delà de cette valeur de C, le score reste maximal de façon constante. Ce résultat est cohérent, car il est attendu qu'un SVM atteigne un score maximal pour une valeur intermédiaire de C. En effet, si C

est trop petit il y a un risque un sous-apprentissage et s'il est trop grand, il y a un risque de sur-apprentissage. Ainsi, le paramètre de régularisation optimal est $C = 0.1$.

```
[0 1 1 1 1 1 1 0 0 1 1 0 1 0 0 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 0 0 1 0 1 1 1 1
 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 0 0 1 0
 1 0 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1
 0 0 1 1 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 0
 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0
 1 1 1 1 1]
[0 1 1 1 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 1 1 1 0 0 1 0 1 1 1 0 0 1 0 1 1 1 1
 1 0 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 1 0 1 1 1 1 0 1 0 1 1 1 1 1 0 0 0 1 1 0 0 0
 1 0 1 1 0 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1
 0 0 1 1 0 1 1 0 1 1 0 1 0 1 1 0 0 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 1 0 1 0
 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0
 1 0 1 1 1]
done in 3.059s
Chance level : 0.6210526315789474
Accuracy : 0.8894736842105263
```

Figure 7: Visualisation de la prédiction, du test et de la précision

Sur la figure7, on observe d'abord la prédiction et le test associé. Nous obtenons une excellente précision de la prédiction (88.9%) avec le classifieur optimal. On a, en figure**??**, un échantillon des résultats obtenues:sur 12 images, 11 ont été correctement prédites.

Figure 8: Prédictions faites à partir du classifieur optimal sur un échantillon d'images

4.2 Variable de nuisances

Montrons que le score de prédiction est sensible au bruit. Pour cela, nous ajoutons des variables de bruit et recommençons la prédiction. Le bruitage consiste à ajouter des variables de bruit (ici 300), et à les mélanger aux autres.
Nous obtenons alors les valeurs suivantes avec et sans nuisances :

```
Score avec variable de nuisance
Generalization score for linear kernel: 1.0, 0.5526315789473685
```

Figure 9: Score avec variable de nuisance

```
Score sans variable de nuisance
Generalization score for linear kernel: 1.0, 0.9105263157894737
```

Figure 10: Score sans variable de nuisance

Nous constatons que le bruit endommage fortement le classifieur qui voit son score presque réduit de moitié.

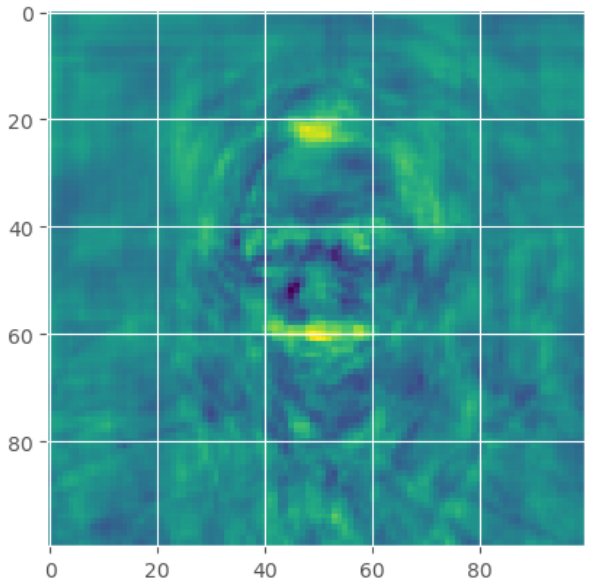


Figure 11: WTF

4.3 Réduction de dimensions

A présent, nous voulons améliorer la prédiction obtenue dans la question précédente à l'aide d'une réduction de dimension.

```
Score apres reduction de dimension
Generalization score for linear kernel: 0.8315789473684211, 0.5263157894736842
```

Figure 12