TP : Support Vector Regression (SVR)

Youssef SALMAN

Objectifs du TP

— Découvrir la mise en œuvre pratique de la SVR en Python (scikit-learn).
— Visualiser le tube ε et les vecteurs de support.
— Étudier l'influence des hyperparamètres C, ε et du noyau.

Dans ce TP, on utilise principalement les bibliothèques Python suivantes :

— `numpy`, `matplotlib`
— `scikit-learn` : `SVR`, `train_test_split`

Partie 1 : Génération et visualisation des données

On s'intéresse à un phénomène (fictif) modélisé par :

$$y = f(x) = \sin(x) + 0{,}3x,$$

auquel on ajoute un bruit gaussien.

1.a) Écrire un code Python qui :
- — génère $n = 80$ points x uniformément répartis sur l'intervalle $[0, 10]$,
- — calcule les valeurs "vraies" $f(x)$,
- — ajoute un bruit gaussien de moyenne 0 et d'écart-type 0,5 pour obtenir y.

1.b) Tracer sur une même figure :
- — la courbe **sans bruit** $f(x)$,
- — les points observés (x_i, y_i) (nuage de points).

1.c) Séparer les données en un **jeu d'apprentissage** (70%) et un **jeu de test** (30%) à l'aide de `train_test_split`.

Partie 2 : Régression linéaire vs SVR linéaire

2.a) Ajuster un modèle de **régression linéaire** classique (moindres carrés) sur le jeu d'apprentissage, tracer la droite prédite sur l'intervalle $[0, 10]$ et commenter la qualité de l'ajustement.

2.b) Ajuster un modèle de **SVR linéaire** (noyau `'linear'`) avec des paramètres par défaut (`C=1.0`, `epsilon=0.1`) et tracer :

— la fonction prédite $\hat{f}(x)$,

— les deux bornes $\hat{f}(x) \pm \varepsilon$.

2.c) Sur le même graphique, superposer :

— les données d'apprentissage,

— la prédiction de la régression linéaire,

— la prédiction de la SVR linéaire,

— le tube ε autour de la SVR.

Commentez les différences visuelles entre les deux approches.

Partie 3 : Vecteurs de support et interprétation

3.a) À partir de l'objet SVR (linéaire), récupérer :

— les `support_vectors_`,

— leurs indices `support_`,

— le nombre total de vecteurs de support.

3.b) Sur le nuage de points, mettre en évidence (par une couleur ou un symbole distinct) les vecteurs de support.

3.c) Interpréter le rôle de ces vecteurs de support en régression SVR : quelles observations influencent directement le modèle ?

Partie 4 : Influence de C et de ε

On fixe le noyau linéaire et on fait varier les hyperparamètres.

4.a) Faire varier C dans l'ensemble $\{0{,}1,\ 1,\ 10,\ 100\}$ avec $\varepsilon = 0{,}1$ fixé. Pour chaque valeur de C :

— ajuster un modèle SVR,

— tracer la prédiction résultante,

— calculer l'erreur RMS sur le jeu de test.

4.b) Faire varier ε dans l'ensemble $\{0{,}05,\ 0{,}2,\ 0{,}5\}$ avec $C = 10$ fixé. Pour chaque valeur de ε :

— ajuster un modèle SVR,

— tracer la prédiction,

— calculer l'erreur RMS sur le jeu de test.

4.c) Résumer dans un tableau l'influence de C et de ε sur :

— la forme du modèle (plus ou moins flexible),

— le nombre de vecteurs de support,

— la performance sur le jeu de test.

Partie 5 : SVR non linéaire (noyau RBF)

5.a) Ajuster un modèle de SVR avec noyau RBF (`kernel='rbf'`) pour quelques choix de paramètres, par exemple :

$$(C, \varepsilon, \gamma) \in \{(10, 0{,}1, 0{,}1),\ (10, 0{,}1, 1),\ (100, 0{,}1, 1)\}.$$

5.b) Pour chaque configuration, tracer la courbe prédite $\hat{f}(x)$ sur $[0, 10]$ et comparer à :

- la fonction vraie $f(x)$,
- la SVR linéaire.

5.c) Discuter l'effet de γ (largeur du noyau RBF) sur la forme de la fonction estimée : quand la courbe devient-elle trop "wiggly" (sur-apprentissage) ?

Partie 6 : Bilan

6.a) En vous appuyant sur les tracés et les erreurs de test, dans quels cas la SVR apporte-t-elle un gain significatif par rapport à la régression linéaire simple ?

6.b) Discuter brièvement :

- le compromis biais–variance en fonction de C, ε et γ,
- l'intérêt des noyaux pour modéliser des relations non linéaires.