Big Data

TD 1 : Support Vector Machine
BUT 3

Guillaume Metzler
Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France
guillaume.metzler@univ-lyon2.fr

Ce premier TD est consacré à la mise en oeuvre des modèles de SVM (*Support Vector Machine*) présentés en cours. L'implémentation se fera à l'aide la librairie `SVC` de Python.
Nous chercherons à mettre en oeuvre les principes de l'apprentissage sur un ensemble de jeu de données afin de comparer les performances des SVM linéaires ou non sur ces données.

1 Mise en pratique

L'objectif est de mettre en oeuvre l'algorithme des SVM sur des jeux de données de la communauté.

1.1 Application

Dans cette première partie, nous allons implémenter l'algorithme des séparateurs à vaste marge sur des jeux de données présentant des caractéristiques différentes. On va considérer les jeux de données : Balance, Pima, Satimage, Segmentation, Yeast6.

Pour ces différents jeux de données, vous allez apprendre, en utilisant une 5-CV

- un SVM avec noyau linéaire en tunant l'hyper-paramètre C parmi les valeurs $\{0.1, 0.5, 1, 2, 4\}$

- un SVM avec un noyau gaussien (ou *radial* sous Python) en tunant l'hyper-paramètre C parmi les valeurs $\{0.1, 0.5, 1, 2, 4\}$ et le paramètre γ parmi les valeurs $\{0.01, 0.1, 1, 10\}$

On souhaite ensuite étudier les performances de ces différents algorithmes. Pour cela, on utilisera le critère de l'*accuracy* défini par

$$\frac{TP + TN}{m},$$

où m représente le nombre d'exemples dans notre jeu de données.

1. Observer les performances sur les différents jeux de données à l'aide de ces deux deux algorithmes. Qu'observez-vous ?

On va maintenant se concentrer sur les jeux de données Satimage, Segmentation et Yeast6 en se concentrant sur le SVM linéaire avec l'hyper-paramètre C égal à 1.

3. Que peut-on dire du taux de classification sur la classe positive et sur la classe négative pour ces trois jeux de données ? On pourra par exemple étudier les matrices de confusion pour ces trois jeux de données.

4. Utiliser le paramètre `class_weight` avec l'option "balanced". Cela permet de modifier le poids des classes lorsque les jeux de données est déséquilibré, de sorte à ce que les classes en présence aient le même poids. Comparer les nouvelles matrices de confusions à celles obtenues à la question précédente.

5. A l'aide des options `dual_support_` et `support_`, comparer le nombre d'exemples de votre jeu de données, à la dimension de ces deux objets. Que remarquez vous ?

6. En vous basant sur la relation permettant de prédire l'étiquette d'une donnée

$$h(\mathbf{x}') = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}', \mathbf{x}_i)\right).$$

et de l'aide concernant la fonction `SVC` et l'output de l'option `dual_support_`, quels sont les exemples qui participent à la construction de la frontière de décision ?

1.2 Evaluation du temps de calcul

L'objectif de cette section est d'évaluer le temps d'apprentissage d'un SVM avec un noyau linéaire et gaussien afin de voir quelles sont les limites de cet algorithme.

Pour cela, on va considérer un jeu de données synthétiques que l'on pourra générer à partir du code qui se trouve au lien suivant

Jeux de données

Les performances des modèles ont peu d'importance dans cette section, on va simplement étudier le temps d'apprentissage, on fixera donc $C = 1$ et $\gamma = 1$ pour les versions linéaire et gaussien des méthodes à noyaux.

1. Représenter, sur un graphique le temps d'apprentissage d'un SVM linéaire et/ou gaussien en fonction de la taille de l'échantillon.

2. Est-ce que cet algorithme est adapté dans un contexte de données massives ? Pourquoi ? Justifier en vous basant sur vos connaissances sur ces méthodes là.

3. Quelles méthodes vues en cours, ou parmi vos connaissances personnelles, pourriez vous employer pour réduire ce temps d'apprentissage.

2 Quelques exercices d'applications

Cette section regroupe quelques exercices d'applications sur les SVM afin de tester la compréhension de ces modèles.

Exercice 1

On considère le jeu de données étiquetées suivant :

y	-1	-1	-1	-1	+1	+1	+1
x_1	2	4	-1	0	1	6	5
x_2	1	3	4	7	-6	-3	-5
α	0.1	0.3	0	0.5	0.7	0	0.2

On souhaite déterminer l'étiquette de la donnée $\mathbf{x}'$ définie par $\mathbf{x}' = (1, 5)$ et on considère un noyau **linéaire**.

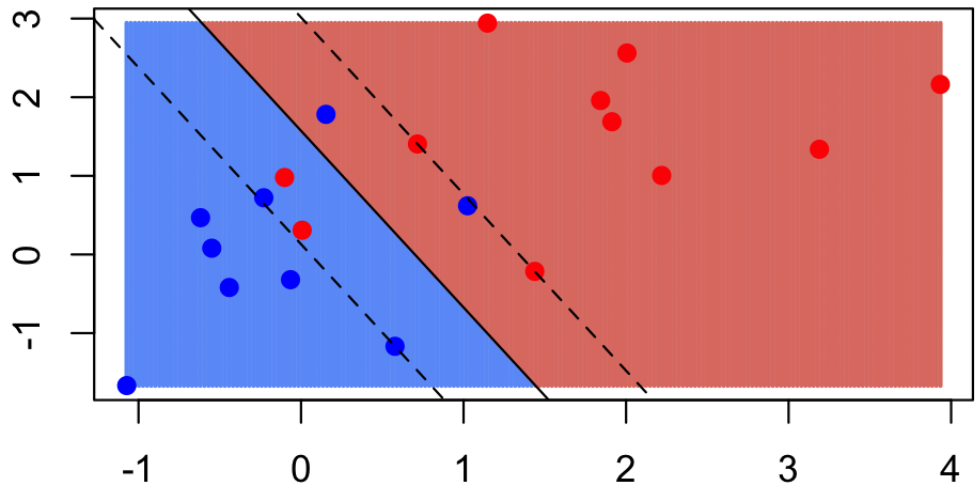

FIGURE 1 – Classifieur SVM linéaire. La zone bleue représente la zone de prédiction négative, *i.e.* $y = -1$ et la zone rouge représente la zone de prédiction positive, *i.e.* $y = +1$.

1. Rappeler la règle de classification pour un SVM à noyaux.
2. Déterminer l'étiquette prédite par le modèle pour la donnée $\mathbf{x}'$ précédemment définie.

Exercice 2

Cet exercice se concentre sur l'étude de certains algorithmes, on va considérer que l'on dispose du jeu d'entraînement S suivant

Individu	x_1	x_2	y
1	-2	1	1
2	1	-1	1
3	3	4	1
4	0	-3	-1
5	1	-2	-1
6	-1	-4	-1

On considère un jeu d'entraînement qui a conduit à l'obtention du SVM linéaire représenté en Figure 1 et dont les paramètres sont approximativement les suivants :

$$\mathbf{w} = (-1.5, -0.5) \quad \text{et} \quad b = -1.$$

1. Rappeler la règle de classification d'un SVM linéaire.

2. Donner la définition de la *hinge loss* et énoncer le problème d'optimisation à résoudre pour un SVM linéaire.
3. Sur la Figure 1, identifier :
   (a) l'hyperplan séparateur
   (b) les marges du SVM
   (c) les vecteurs (ou points) supports
4. Prédire l'étiquette des individus $\mathbf{x}_1, \mathbf{x}_2$ et $\mathbf{x}_6$ à l'aide du SVM (on demande de le faire par un calcul et non graphiquement).
5. Quelle est la valeur de la loss pour le point $\mathbf{x}'$ de coordonnées $(1, -1)$ qui est un point dont le label est négatif, *i.e.* $y = -1$.