TP – Support Vector Machines & Decision trees (Part 2)

C. SVM non linéaire à noyau gaussien

1) Générer une base de données B3 contenant 200 exemples, à l'aide de la fonction *makemoons*, en fixant le bruit à un niveau faible (noise=0.1). Séparer en bases d'apprentissage et de test.

2) Analyser l'influence du paramètre de dispersion (spread) γ du noyau gaussien.

   a) Définir un SVM à noyau gaussien de paramètre γ, en fixant la valeur de C.

   b) Faire varier γ (échelle linéaire). Pour chaque valeur de γ, observer les frontières, calculer les erreurs en apprentissage et en test, noter le nombre de vecteurs supports et les afficher. Rassembler les résultats dans un tableau. Conclure en termes de biais et de variance du modèle.

   c) Conclure.

3) Répéter les expériences 2.A et 2.b en générant une base de données B4 en fixant le bruit à un niveau plus élevé (noise=0.3). Conclure.

4) Optimisation par recherche en grille (*grid search*)

   a) Faire varier γ en même temps que C. Calculer l'erreur et la matrice de confusion sur les bases d'apprentissage et de test. Afficher les frontières de décision et les analyser. Noter le nombre de vecteurs supports. Rassembler tous les résultats dans un tableau.

   b) Répéter le processus de recherche en grille en effectuant une validation croisée à 5-plis (5-fold cross validation). Conclure.

D. Arbres de décision

On utilisera la base de données IRIS (https://archive.ics.uci.edu/ml/datasets/iris) et on choisira deux caractéristiques parmi les 4 pour pouvoir observer les résultats dans un plan.

1) Séparer la base de données du TP1 en deux : apprentissage et test (échantillonnage stratifié sans remise).

2) Définir un arbre de décision en utilisant comme critère l'entropie (*criterion='entropy'*) et analyser l'influence des paramètres suivants sur les performances en apprentissage, en test et sur les frontières de décision.

   a) Entraîner l'arbre
   b) Faire varier *max_depth* et noter les performances
   c) Faire varier *min_samples_split* et noter les performances
   d) Conclure en termes de biais et de variance

3) Trouver les valeurs optimales de *max_depth* et *min_samples_split*, *via* une recherche en grille (*gridsearch*) en effectuant une validation croisée à 5-plis (*5-fold cross validation*). Estimer les performances sur la base de test.