Objectif

Utiliser la méthode C-SVC (SVM pour l'apprentissage supervisé multi-classes) en provenance de la bibliothèque LIBSVM[1].

LIBSVM est une bibliothèque qui regroupe un ensemble d'algorithmes de fouille de données (classement, régression…), toutes issues de l'approche par les machines à support de vecteurs (Support Vector Machine). L'implémentation est particulièrement efficace, particulièrement en ce qui concerne le temps de traitement comme nous le verrons plus bas. Il est également possible d'accéder à une série de documentations sur le site web des auteurs de la bibliothèque.

Le code source en C est disponible en ligne, nous nous sommes contentés de la compiler en DLL puis de l'associer aux composants de TANAGRA. Dans un premier temps, seule C-SVC a été introduite. A terme, d'autres méthodes, telles que la régression, seront également intégrées.

Fichier

Un fichier issu de la discrimination de protéines à partir de leurs structures primaires (Mhamdi et al., 2004).

Le fichier contient 122 individus répartis en deux familles de protéines {C1, C2}, il y a 6740 descripteurs booléens (1/0) qui correspondent à la présence/absence de 3-grams extraits de la description « brute » des données.

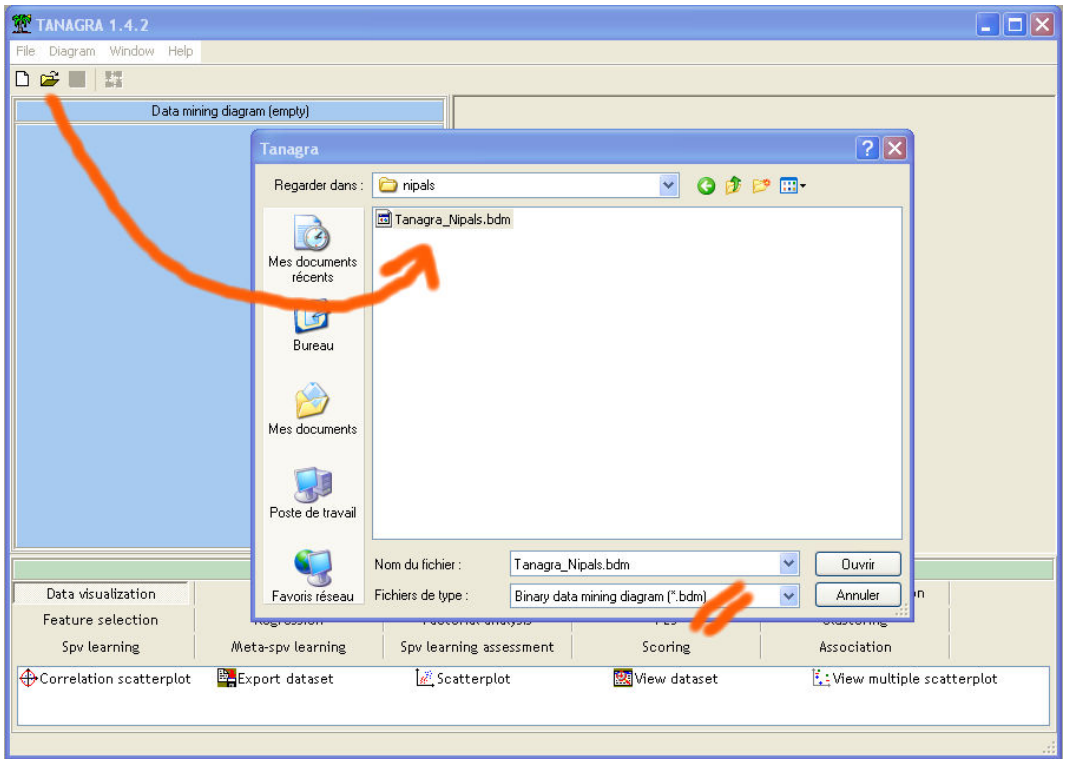
Ce fichier a déjà été utilisé pour montrer l'intérêt de la construction de variables intermédiaires lorsque la dimension de représentation est très élevée. Il s'agissait de produire des axes factoriels à l'aide l'analyse en composantes principales (NIPALS), puis d'utiliser la méthode des plus-proches voisins dans le nouvel espace de représentation.

C-SVC

Charger le fichier de données

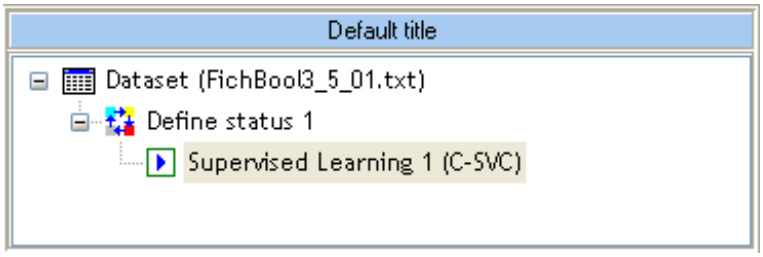
Charger le fichier TANAGRA_NIPALS.BDM. Attention, ce fichier de données est au format binaire, il est donc nécessaire de choisir le format « Binary Data Mining Diagram » dans la boîte de saisie du nom de fichier.

[1] Disponible à l'adresse http://www.csie.ntu.edu.tw/~cjlin/libsvm/



Apprentissage supervisé

Essayons de produire et d'évaluer la méthode C-SVC issue de la bibliothèque LIBSVM. Il faut pour ce faire : sélectionner les attributs TARGET (*Classe*) et INPUT (*tous les autres descripteurs*), placer les composants d'apprentissage. Le diagramme de traitements est le suivant.



Le paramétrage par défaut correspond à un SVM linéaire. Nous constatons qu'une séparation parfaite a été trouvée, ce qui n'est pas étonnant étant donné le ratio nombre de descripteurs (6740) et nombre d'observations (122). Plus intéressant en revanche est le temps de traitement qui est particulièrement avantageux, 2.7 secondes sur un P4 à 3Ghz, le passage des données vers la DLL se révèle très peu coûteux ; il semble également que la bibliothèque a été élaborée avec beaucoup de soin.

Classifier performances

Error rate	0.0000
Values prediction	Confusion matrix

Value	Recall	1-Precision
C1	1.0000	0.0000
C2	1.0000	0.0000

	C1	C2	Sum
C1	54	0	54
C2	0	68	68
Sum	54	68	122

Classifier characteristics

Data description

Target attribute	Classe (2 values)
# descriptors	6740

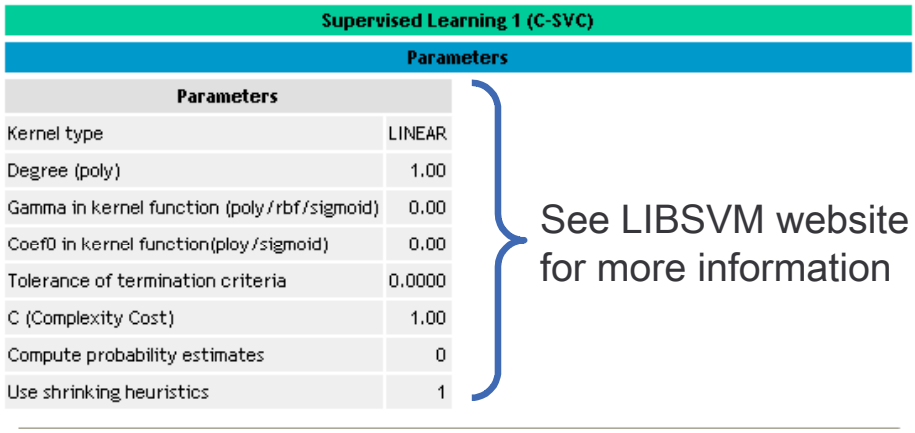
SVM characteristics

Characteristic	Value
# classes	2
# support vectors	97
# support vectors for each class	
# sv. for C1	44
# sv. for C2	53

Computation time : 2672 ms.
Created at 09/01/2006 16:16:20

Le composant affiche la matrice de confusion, les caractéristiques de l'apprentissage, notamment le nombre de points supports pour chaque modalité de la variable à prédire.

Les paramètres de la méthode utilisée sont listés sur la partie haute du rapport. Nous vérifions bien qu'un SVM linéaire a été utilisé (KERNEL TYPE).



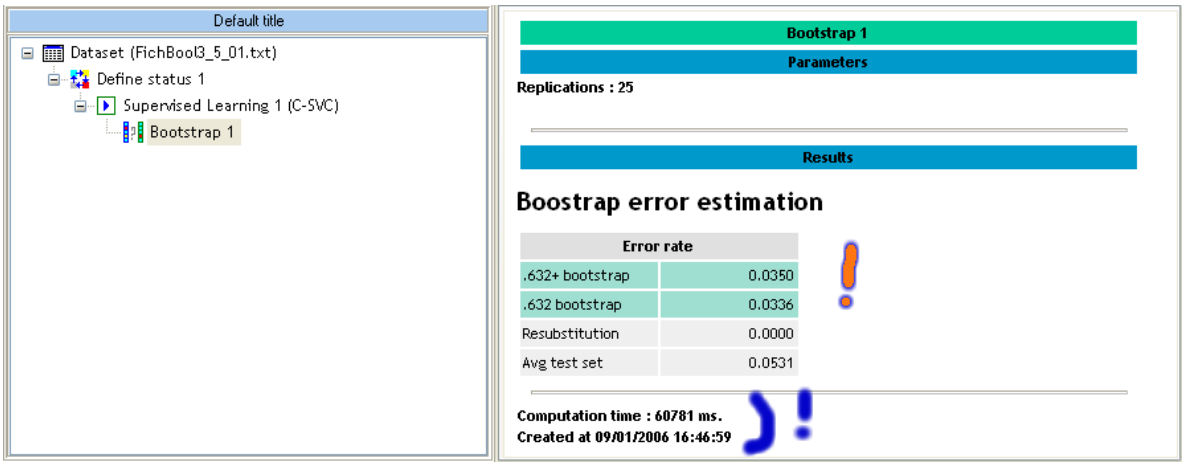
Supervised Learning 1 (C-SVC)

Parameters

Parameters	
Kernel type	LINEAR
Degree (poly)	1.00
Gamma in kernel function (poly/rbf/sigmoid)	0.00
Coef0 in kernel function(ploy/sigmoid)	0.00
Tolerance of termination criteria	0.0000
C (Complexity Cost)	1.00
Compute probability estimates	0
Use shrinking heuristics	1

Evaluation

Le taux d'erreur en apprentissage est très souvent biaisé. Nous devons donc utiliser une méthode de ré-échantillonnage pour obtenir une évaluation plus fiable de l'erreur : nous avons utilisé la méthode « Bootstrap plus » (Efron & Tibshirani, 1997) disponible dans la palette SPV LEARNING ASSESMENT. Le diagramme a été complété de la manière suivante.



Le taux d'erreur estimé est de 3.5% et le temps de calcul total est de 60 secondes. Ces résultats montrent encore une fois combien les SVM, surtout linéaires, par la conjugaison de du biais de représentation et du biais d'apprentissage, résistent au sur-apprentissage, même lorsqu'ils sont placés dans ces conditions extrêmes de surdimensionnalité où l'on a 50 fois plus de descripteurs que d'observations.

Ces résultats prennent toutes leurs saveurs lorsque l'on se remémore les performances des K-plus proches voisins (K-NN) placés dans les mêmes conditions.