TP1 - Apprentissage supervisé

Paul Chaignon - Ulysse Goarant

6 février 2014

1 Apprentissage d'un SVM

1.1 Données linéairement séparables

Étape 1

Les deux classes peuvent être séparées ici par une infinité de droites, ici deux possibles sont représentées.

Étape 2

La droite optimale a pour équation : y = 1.77x − 0.88. Le risque empirique est nul (tous les éléments appartenant à l'ensemble d'apprentissage sont bien classés).

Étape 3

En séparant l'ensemble des exemples en un ensemble d'apprentissage (50%) et un ensemble de test (50%), le risque réel est nul. Cependant, si l'on utilise seulement 10% des données comme ensemble d'apprentissage, le risque réel croit à 55%.

1.2 Données non linéairement séparables

En utilisant un noyau Puk, il est possible de classer des exemples non linéairement indépendants. Il y a dans ce cas 137 vecteurs de support impliqués.

2 Apprentissage d'un arbre décision

2.1 Construction et évaluation d'arbres

Étape 1

Le fichier *weather.nominal.arff* contient 14 instances. Ils ont chacun 5 attributs dont 4 de type nominal et 1 de type booléen. La classe à prédire est « play ».

Étape 2 & 3

40% des données de test ont bien été classées. La matrice de confusion nous indique que le classement a été plus efficace à classer les « yes ».

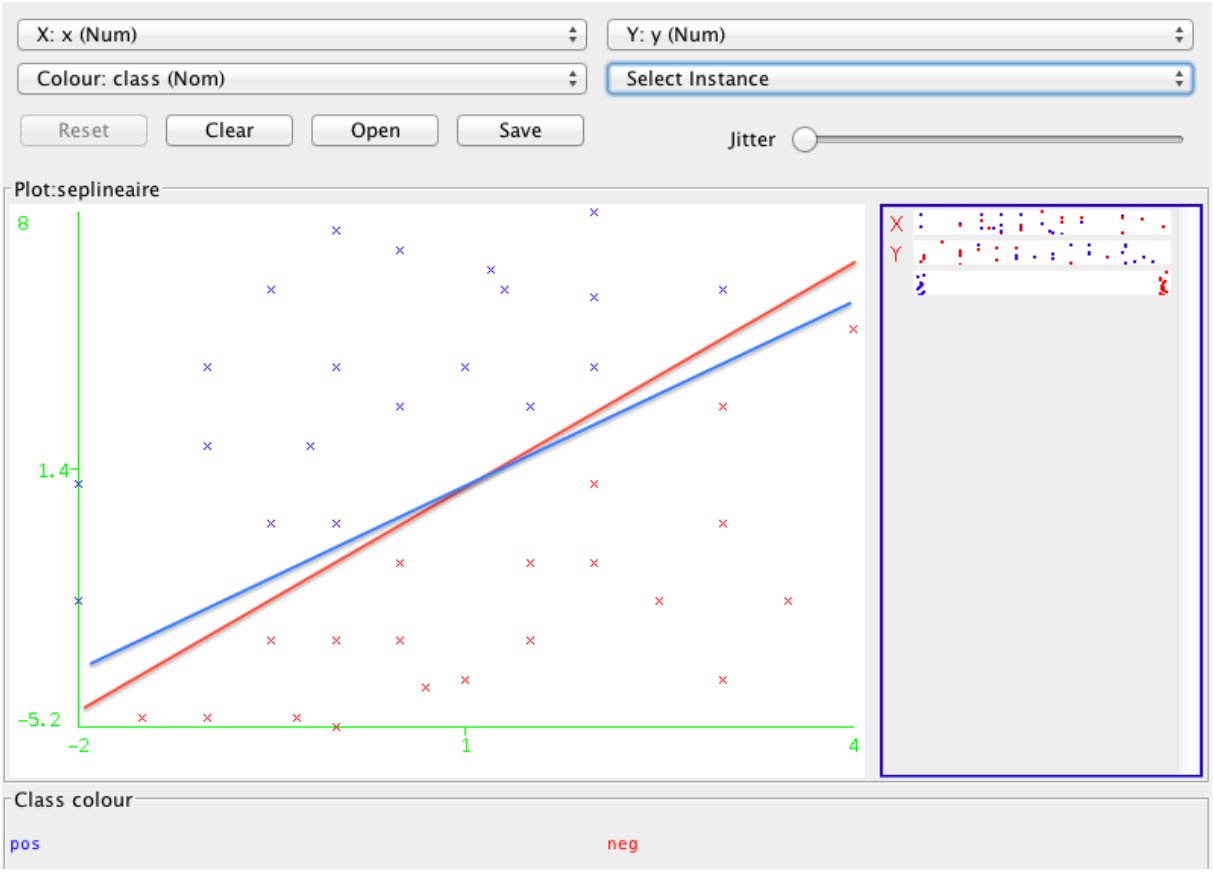

FIGURE 1 – Données linéairement séparables

Text

```
=== Evaluation result ===

Scheme: SMO
Options: -C 1.0 -L 0.001 -P 1.0E-12 -N 2 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel
Relation: seplineaire-weka.filters.unsupervised.attribute.ClassAssigner-Clast


Correctly Classified Instances          20               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances               20

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        pos
                 1         0          1         1         1          1        neg
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

  a  b   <-- classified as
 12  0 |  a = pos
  0  8 |  b = neg
```

FIGURE 2 – Résultat de la méthode SMO sur les données linéairement séparables

```
=== Evaluation result ===

Scheme: J48
Options: -C 0.25 -M 2
Relation: weather.symbolic-weka.filters.unsupervised.attribute.ClassAssigner-Clast


Correctly Classified Instances           2               40      %
Incorrectly Classified Instances         3               60      %
Kappa statistic                         -0.3636
Mean absolute error                      0.6
Root mean squared error                  0.7746
Relative absolute error                126.9231 %
Root relative squared error            157.6801 %
Total Number of Instances                5

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.667     1          0.5       0.667     0.571      0.333    yes
                 0         0.333      0         0         0          0.333    no
Weighted Avg.    0.4       0.733      0.3       0.4       0.343      0.333

=== Confusion Matrix ===

 a b   <-- classified as
 2 1 | a = yes
 2 0 | b = no
```

FIGURE 3 – Arbre de décision J48

```
=== Evaluation result ===

Scheme: J48
Options: -C 0.25 -M 2
Relation: weather.symbolic-weka.filters.unsupervised.attribute.ClassAssigner-Clast


Correctly Classified Instances           6               75      %
Incorrectly Classified Instances         2               25      %
Kappa statistic                          0
Mean absolute error                      0.5
Root mean squared error                  0.5
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances                8

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.75      1         0.857      0.5      yes
                 0         0          0         0         0          0.5      no
Weighted Avg.    0.75      0.75       0.563     0.75      0.643      0.5

=== Confusion Matrix ===

 a b   <-- classified as
 6 0 | a = yes
 2 0 | b = no
```

FIGURE 4 – Amélioration de l'arbre de décision J48

Étape 4

En réduisant la part de l'ensemble des données d'apprentissage à 40% et en modifiant la graine, le risque réel est réduit à 25%.

Étape 5

Ce nouveau jeu de données contient des attributs numériques. L'arbre construit contient donc des nœuds testant des inégalités.

3 Élagage et simplification

L'arbre non-élagué obtient un meilleur taux de risque réel (40%) comparé à l'arbre élagué (60%), cependant le premier arbre a un plus grand risque d'avoir « coller aux données » que le second et a donc une capacité de généralisation moins forte.

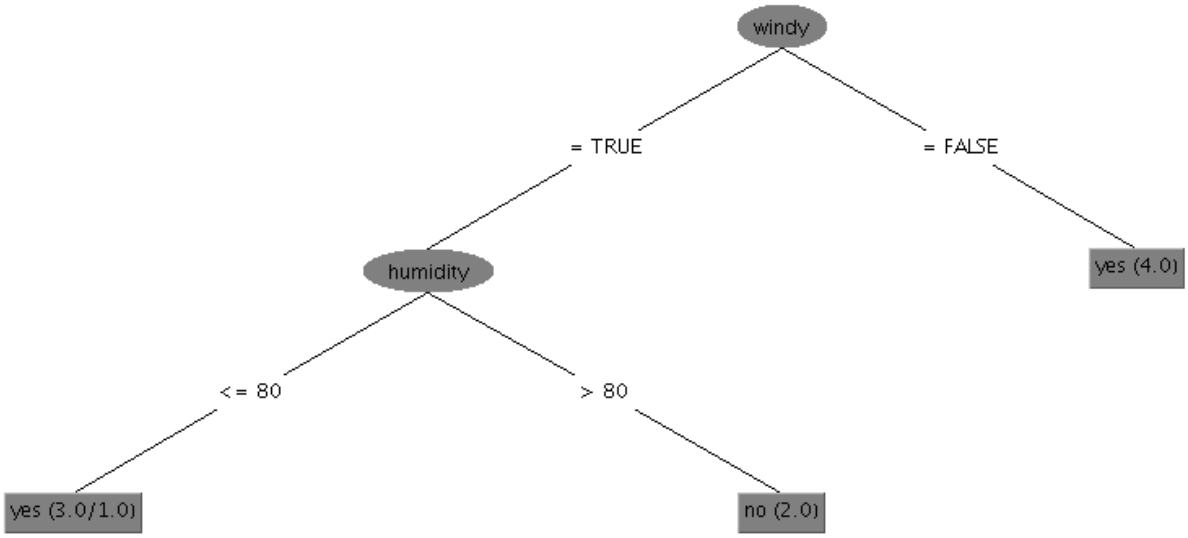

FIGURE 5 – Arbre de décision élagué

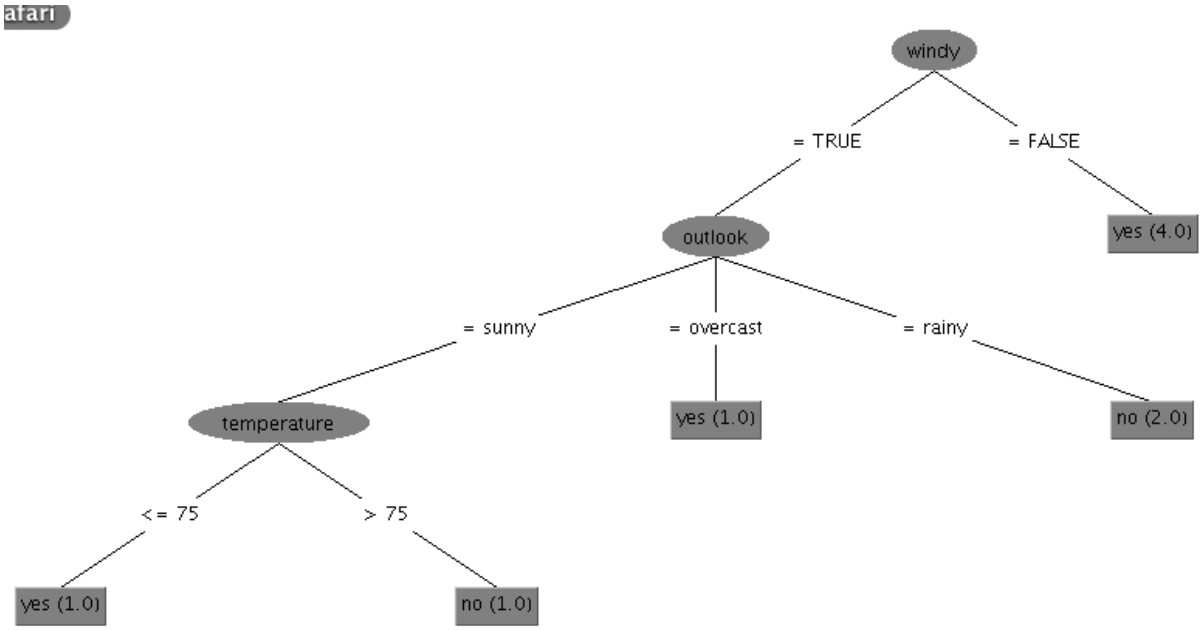

FIGURE 6 – Arbre de décision non élagué

4 Apprentissage bayésien

4.1 Bayes naïf

Étape 1

L'hypothèse ici utilisée est que les attributs n'ont pas d'influence les uns sur les autres.

Étape 2

Pour les attributs numériques que sont *température* et *humidité*, le modèle nous fournit une estimation des paramètres de leurs lois gaussiennes respectives. Pour les attributs nominaux ou booléen (*outlook* et *windy*), l'estimateur de Laplace est utilisé.

Étape 3

Selon Weka, cette exemple est associé au label non. Aussi, l'arbre de décision élagué construit précédemment l'aurait également dans cette catégorie.

Étape 4

A partir des exemples du fichier *weather.arff* découpant en un ensemble d'apprentissage (deux tiers) et un ensemble de test (un tiers), on obtient un taux d'erreur réel de 40% ce qui est comparable avec les résultats obtenus grâce aux arbres.

4.2 Approche non paramétrique

Étape 1

L'algorithme IBk associe à un exemple à classer le label le plus présent parmi ses plus proches voisins dont le nombre est ici choisi à 1. Le label considéré comme étant le plus présent parmi un ensemble de voisins peut être choisi selon divers critères.

Étape 2

L'algorithme IB1 parvient à classer correctement 96% des exemples de l'ensemble de test.

Étape 3 et 4

L'algorithme 2-NN fournit les mêmes résultats. On peut cependant augmenter d'avantage le nombre de voisins pour les améliorer.