Deuxième Licence en Informatique
Data Warehousing et Data Mining
La Classification - 1

V. Fiolet
Université de Mons-Hainaut
2006 - 2007

Nous allons aujourd'hui nous intéresser à la tâche de classification et à l'utilisation du Weka Experimenter.

1 Détails sur l'évaluation d'une classification

Pour effectuer une classification , différents points doivent être pris en compte:

- Les effets des jeux de tests utilisés (cf tp précédent)
- Les effets du type de validation de modèles :
  - Validation sur le jeu de données ayant servi à l'apprentissage
  - Validation sur un autre jeu de données (sur une deuxième partition des données)
  - Validation croisée

Nous allons au cours de cette séance et de la suivante voir l'influence de ces deux élements sur les résultats de la classification et sur la qualité des solutions obtenues (qualité réelle et mesures fournies par les indicateurs).

But: A la fin de cette séance sur le Weka Experiment Environment vous serez capables de concevoir des expériences pour évaluer les schémas de machine learning des classifieurs sur de multiple datasets.

2 Introduction Environnement

Le Weka Experiment Environment permet à l'utilisateur de créer, lancer, modifier et analyser des expériences de manière plus souple que l'utilisation des classifieurs individuellement. Par exemple, l'utilisateur peut créer une expérience qui lance plusieurs classifieurs sur des séries de datasets et analyse les résultats pour déterminer si l'un des classifieur est statistiquement meilleur que les autres. Nous allons utiliser l'interface de l'Experiment Environment puisqu'il fournit plus de flexibilité à l'utilisateur pour développer des expériences que cela est possible en tapant les commandes dans une simple console (CLI). Pour commencer avec l'interface de l'Experiment Environment, démarrez Weka et cliquez sur Experimenter dans la fenêtre Weka GUI Chooser.

2.1 Définir une Expérience

Quand l' Experimenter est démarré, la fenêtre **Setup** est affichée. La première étape consiste à choisir un dataset de travail.

Exercice 2.1 *Cliquez sur* **New** *(en haut à droite de la fenêtre* **Setup***) pour initialiser une expérience. Ceci permet de fixer les paramètres par défaut pour l'expérience. Pour définir le dataset à traiter par un classifieur, sélectionnez d'abord* **Use relative paths** *dans la fenêtre* **Datasets** *(en bas à gauche de la fenêtre* **Setup***) et cliquez sur* **Add new** *pour ouvrir une fenêtre de dialogue. Double-cliquez sur* **data** *pour voir les datasets disponibles ou naviguez vers un autre répertoire. Sélectionnez le dataset* **Iris***. Le nom du dataset est maintenant affiché dans la fenêtre* **Datasets** *de la fenêtre de* **Setup***.*

Se placer en mode advanced en haut de la fenêtre.

2.2 Sauvegarder les Résultats de l' Expérience

Afin de ne pas refaire le même travail plusieurs fois et de pouvoir exploiter les résultats de l'expérience, on peut sauvegarder les résultats de l'expérience dans un fichier.

Exercice 2.2 *Pour identifier un fichier vers lequel les résultats seront envoyés, cliquez sur* **CSVResultListener** *dans la fenêtre* **Destination** *(en haut de la fenêtre* **Setup***). Se placer sur le paramètre de l'output* **outputFile***. Cliquez sur ce paramètre pour afficher une fenêtre de sélection de fichiers. Tapez le nom du fichier de sortie (output), par exemple Experiment.txt ou le sélectionner via open, cliquez sur* **Select***, et cliquez sur close (X). Vérifiez que tous les fichiers de sortie sont situés dans votre répertoire (et pas dans le répertoire weka). Le nom du fichier est affiché dans la fenêtre* **outputFile***. Cliquez sur OK pour fermer la fenêtre. Le nom du dataset est affiché dans la fenêtre* **Destination** *du* **Setup***.*

2.3 Sauvegarder la définition de l'Expérience

La définition d'une expérience peut être sauvegardée à tout moment.

Exercice 2.3 *Sélectionnez* **Save** *en haut de la fenêtre* **Setup***. Tapez le nom du dataset avec l'extension "exp" (e.g. Experiment.exp).*

Récupération: L'expérience peut être récupérer en sélectionnant **Open** (en haut à gauche de la fenêtre **Setup**) et sélectionnez Experiment.exp dans la fenêtre de dialogue.
Remarque: Pensez à sauvegarder les définitions de vos expériences de manière à pouvoir les réutiliser plus tard.

2.4 Lancer une expérience

Exercice 2.4 *Pour lancer l'expérience que vous venez de configurer, cliquez sur l'onglet* **Run** *en haut de la fenêtre* **Experiment Environment** *et cliquez sur* **Start***. L'expérience est exécutée.*

Dans le cas présent, c'est l'expérience par défaut qui est exécutée sur le dataset que vous avez choisi, soit le CrossValidationResultProducer avec un paramètre à 10 pour l'apprentissage aléatoire et des tests effectués sur le dataset choisi (Iris), utilisant le classifieur *ZeroR*.
Se placer en mode disabled dans le generator properties.

```
Started
Finished
There were 0 errors
```

Si l'expérience a été définie correctement, les 3 messages présentés ci-dessus sont affichés dans le **Log Panel**. Les résultats de l'expérience sont sauvegardés dans le fichier Experiment.txt. C'est une fichier .CSV (valeurs séparées par des ";") qui peut être chargé dans un classeur (Excel ou OpenOffice) pour être analysé.

Exercice 2.5 *Chargez ce fichier dans une feuille Excel. Spécifiez le délimiteur à ";".*

La ligne 1 contient les identifiants de colonnes (du résultat de l'expérience). Chaque ligne définit un jeu d'apprentissage et un test.
La ligne 2 de la feuille de données indique que pour le premier run de l'expérience, le dataset *Iris* a été utilisé avec le classifieur *ZeroR* et que W instances ont été testées par le classifieur: X instances ont été classées correctement, Y instances ont été mal classées, et Z instances n'ont pas pu être classées.

Exercice 2.6 *Laquelle (lesquelles) de ces colonnes pourrai(en)t être utile pour analyser l'efficacité d'un classifieur ?*

2.5 Modifier les paramètres de l'expérience

Les paramètres d'une expérience peuvent être modifiés en cliquant sur la partie **Result Generator** (sous Destination dans la fenêtre de **Setup**). Redimensionnez la fenêtre pour avoir tous les paramètres visibles. Le RandomSplitResultProducer réalise des apprentissages et tests de manière répétée. Le nombre de cas (exprimé sous forme de pourcentage) à utiliser pour l'apprentissage est donné dans le champ **trainPercent**.
Le nombre de runs est spécifié dans la fenêtre **Setup** dans le paramètre **Runs**. Un petit fichier d'aide peut être affiché en cliquant **More** dans le panneau **About**.

Exercice 2.7 *Cliquez sur l'entrée* **splitEvaluator** *pour afficher les propriétés du* **SplitEvaluator**. *Cliquez sur l'entrée classifier (ZeroR) pour afficher les propriétés du classifieur. Ce classifieur n'a pas de propriétés modifiables mais la plupart des autres classifieurs en ont (e.g. j48.J48) et peuvent être modifiés par l'utilisateur. Cliquez sur la liste déroulante pour le classifieur ZeroR (***choose***) et changez le en j48.J48 pour le classifieur arbre de décision.*

Exercice 2.8 *Vous pouvez modifier les paramètres, par exemple augmentez le minNumObj de 2 à 5 (i.e. spécifiez le nombre minimum de cas dans les noeuds feuilles).*

Exercice 2.9 *Cliquez maintenant sur OK pour fermer la fenêtre.*
Le nom du nouveau classifieur est affiché dans le panneau **Result generator**. *Vous pouvez relancer l'expérience. Le fichier d'output (Experiment.txt) sera écrasé par les résultats du j48.J48.*

2.6 Comparer des Classifieurs

Pour comparer plusieurs classifieurs, nous devons les ajouter dans le panneau de **Generator properties**.

Exercice 2.10 *Pour commencer, modifiez l'entrée drop-down de Disabled à Enabled dans le panneau* **Generator properties** *(en bas à droite de la fenêtre* **Setup***). Cliquez sur* **Select property** *et sélectionnez* **splitEvaluator** *de manière à ce que l'entrée classifier soit visible dans la liste des propriétés. Cliquez sur* **Select**.
Le nom du classifieur est affiché dans le panneau **Generator properties**.
Sélectionnez **ZeroR** *comme classifieur. Puis utilisez le bouton* **Add** *pour l'ajouter à l'expérience.*

Pour ajouter un autre classifieur, cliquez sur le nom du classifieur pour afficher sa fenêtre de propriétés. Cliquez sur la liste déroulante et sélectionnez j48.J48 , le classifieur arbre de décision. Le nouveau classifieur est ajouté dans le panneau **Generator properties**. *Cliquez sur* **Add** *pour l'ajouter.*
Lancez l'expérience et regardez le fichier résultat Experiment.txt. Vous verrez que les résultats ont été générés à partir des deux classifieurs. Pour ajouter d'autres classifieurs, répétez le procédé.
Pour enlever un classifieur, sélectionnez le en cliquant dessus et cliquez **Delete**.

2.7 Ajouter des jeux de données (Datasets)

Le(s) classifieur(s) peuvent être lancés sur un nombre quelconque de datasets à la fois. Les datasets sont ajoutés en cliquant sur **Add new** dans le panneau **Datasets** (en bas à gauche de la fenêtre **Setup**). Les datasets sont effacés de l'expérience en sélectionnant le dataset voulu et en cliquant sur **Delete Selected**.

3 Les différentes méthodes de validation de modèle

Il existe différentes méthodes de validation de modèles en Weka.
Dans l'explorer, nous avons utilisé, lors du tp précédent, la validation sur jeu d'apprentissage (*Use training set*). Cette méthode consiste à évaluer la qualité du modèle créé par rapport au jeu de données ayant servi à le construire.
Dans l'explorer toujours, trois autres méthodes de validation étaient proposées:

- *Supplied test set* (avec paramètre set - choix du jeu de données pour la validation): consiste à évaluer le modèle sur un autre jeu de données (a priori différent de celui utilisé pour construire le modèle)
- *Cross-validation* (avec paramètre folds): consiste à diviser les données en n groupes. On construit les modèles sur n-1 groupes et on les teste sur le nième groupe. Puis on change de groupe test et on répète le même procédé jusqu'a avoir réalisé toutes les combinaisons. On considère alors la moyenne des validations comme la validation finale.
- *Percentage split* (avec paramètre pourcentage): consiste à utiliser un certain pourcentage des données pour construire le modèle et l'autre partie pour le valider.

4 Expériences

4.1 Premier Pas

4.1.1 Exercices avec le RandomSplitResultProducer

Exercice 4.1 *Définissez une expérience respectant les règles suivantes:*

- *appliquant 10 runs répétitifs d'apprentissage /test*

- *utilisant les datasets Iris et Soybean*
- *avec ZeroR, OneR et j48.J48*

Lisez les résultats dans les fichiers output .CSV (Experiment1.txt) dans Excel. Souvenez vous de sauvegarder la définition de votre expérience (Experiment1.exp) et assurez-vous que tous les fichiers d'output sont sauvegardés dans votre répertoire.
Pour chacun des classifieurs, sur l'un des datasets de votre choix, utilisez les fonctions Excel pour calculer:

- *la moyenne du* **Percent correct** *sur 10 runs ;*
- *l'écart type (SD), toujours pour la colonne* **Percent correct** *sur 10 runs;*
- *l'interval de confiance (CI) à 95% de confiance (indication: alpha 0.05 et la valeur de SD sont nécessaires).*

Utilisez les fonctions excel **INTERVALLE.CONFIANCE** *et* **ECARTTYPE** *Stockez les valeurs de moyenne et de CI values dans des cellules séparées.*
Par exemple:

```
-----------------------------
AVG      ZeroR    OneR   j48J48
Iris     28.9     ...    ...
Soybean  11.6     ...    ...

CI       ZeroR    OneR   j48J48
Iris     1.9      ...    ...
Soybean  1.09     ...    ...
-----------------------------
```

Créez un graphique Excel des résultats de moyenne. Un exemple d'un tel graphique est donné Figure 1. Double cliquez sur une série (dans le graphique) pour ajouter une barre d'erreur Y. Cliquez sur **Custom** *et sélectionnez les cellules significatives d'intervalle de confiance pour les valeurs + et -.* **Quelles conclusions peuvent être déduites à propos des performances des classifieurs sur les différents datasets?**

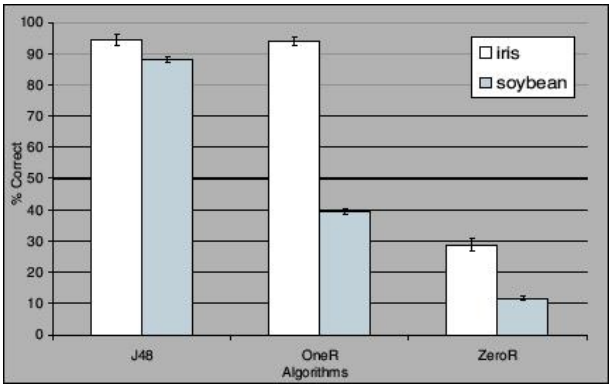

Figure 1: Comparaison de 3 classifieurs sur 2 datasets.

4.2 Expériences avec la validation croisée (Cross-Validation)

Exercice 4.2 *Pour passer d'apprentissage et des tests aléatoires (comme dans l'exercice précédent) à des expériences par validation croisée, cliquez sur l'entrée* **Result generator**. *Cliquez sur la liste déroulante en haut de la fenêtre et sélectionnez* **CrossValidationResultProducer**.

La fenêtre contient maintenant les paramètres spécifiques pour la validation croisée tels que le nombres de partitions (*folds*). L'expérience par défaut travaille avec une validation croisée sur 10 partitions. Le panneau **Result generator** indique maintenant que la validation croisée sera utilisée. Vous pouvez cliquer sur **More** pour générer de brefs descriptions du producteur de résultats par validation croisée. Tout comme le **RandomSplitResultProducer**, de nombreux classifieurs peuvent être lancés pendant la validation croisée en les ajoutant au panneau **Generator properties**.

4.2.1 Exercices avec CrossValidationResultProducer

Exercice 4.3 *Définissez une expérience respectant les règles suivantes:*

- *10 runs de 10 partitions pour la validation croisée ;*
- *utilisant 3 datasets (e.g. Iris, Labor, et Weather) ;*
- *avec IBk (avec KNN fixé à 3) et j48.J48.*

Sauvegarder les résultats de sortie dans un fichier .CSV (Experiment2.txt) et lisez ce fichier avec Excel.

Exercice 4.4 *Réalisez un graphique avec Excel comme dans l'exercice précédent (Section 4.1).*

Notez que 600 (10 runs 10 partitions 3 datasets 2 classifieurs) lignes de résultats ont été générées. Pour effectuer l'analyse, vous devez d'abord faire la moyenne sur les partitions avant de pouvoir faire les moyennes sur les runs.

4.3 AveragingResultProducer

Nous allons utiliser en alternative au **CrossValidationResultProducer**, le **AveragingResultProducer**.

Ce "producer" de résultats fournit la moyenne d'un jeu de tests (généralement des runs de validation croisée).

Exercice 4.5 *Cliquez sur le panneau* **Result Generator** *et sélectionnez* **AveragingResultProducer** *dans la liste déroulante.*

Tout comme pour les autres "producers", d'autres classifieurs peuvent être définis. Quand le **AveragingResultProducer** est utilisé, la propriété *classifier* est située plus loin dans la hiérarchie de **Generator properties**. Avec *expectedResultsPerAverage* fixé à 10, l'expérience consistera en 10 runs de 10 partitions de validation croisée. Chaque run de 10 partitions de validation croisée est alors moyenné, produisant une ligne de résultat pour chaque run (au lieu d'un résultat par ligne pour chaque partition dans l'exemple précédent utilisant le **CrossValidationResultProducer**).
Si un fichier d'output différent est précisé, tous les résultats individuels (sans moyenne) sont envoyés dans une archive pre-spécifiée.

4.3.1 Exercices avec le AveragingResultProducer

Exercice 4.6 *Répétez l'exercice de la Section Cross-Validation 4.2 en utilisant maintenant le* **AveragingResultProducer***, sauvergardez les résultats sous forme CSV (Experiment3.txt) et lisez ce fichier dans Excel.*

Notez qu'il y a seulement 60 lignes de résultats au lieu de 600.

Exercice 4.7 *Pour chacun des classifieurs :*

- *Créez un graphique (similaire à la Figure 1) et comparez les graphiques pour la moyenne de Percent correct.*
  Y-a-t-il des différences significatives en relation avec l'effectivité?
- *Créez d'autres graphiques mais cette fois-ci pour comparer les temps d'apprentissages et de test.*
  Y-a-t-il des conclusions flagrantes qui peuvent être tirées en relation avec l'efficacité?

Deuxième Licence en Informatique
Data Warehousing et Data Mining
La Classification - 2

V. Fiolet
Université de Mons-Hainaut
2006 - 2007

Cette séance poursuit l'étude du **Weka Experiment Environment** et se focalise sur la manière dont le **Weka Experiment Analyser** peut être utilisé pour analyser les résultats.

1 Analyser les resultats avec Weka

En plus d'envoyer les résultats d'une expérience dans un **CSV Result Listener** (fichier d'output voir tp précédent), ces résultats peuvent également être envoyés à un **InstancesResultListener** et analyser par le **Weka Experiment Analyser**. Ceci permettra de ne pas utiliser un logiciel tel qu'excel (cf tp précédent) pour l'analyse des résultats.

Exercice 1.1 *Cliquez sur la partie* **Result Listener** *du panneau* **Destination** *et sélectionnez* **InstancesResultListener**.

N'oubliez pas de vous placer en mode advanced (en haut de la fenêtre).
Spécifiez le nom du result output. L'output sera sous un format dataset, donc spécifiez l'extension "arff" (e.g. Experiment4.arff).

Une fois encore assurez-vous que les fichiers d'output sont enregistrés dans votre répertoire personnel.
L'output créé avec le **InstancesResultListener** est dans un format "arff" de type:

```
@relation InstanceResultListener
@attribute Key_Dataset {iris}
@attribute Key_Run {1,2,3,4,5,6,7,8,9,10}
@attribute Key_Scheme {weka.classifiers.ZeroR}
@attribute Key_Scheme_options {''}
@attribute Key_Scheme_version_ID {6077547173920530258}
@attribute Date_time numeric
@attribute Number_of_instances numeric
@attribute Number_correct numeric
@attribute Number_incorrect numeric
@attribute Number_unclassified numeric
@attribute Percent_correct numeric...
@data
```

Chacunes des instances de ce dataset représente les informations relatives à un run (une ligne dans les tables excel utilisées précédemment).

Le **Weka Experiment Analyzer** peut maintenant être utilisé pour effectuer le travail d'analyse des résultats des expériences (envoyés à un **InstancesResultListener**).

Exercice 1.2 *Ajoutez 3 classifieurs* **ZeroR**, **OneR** *et* **j48.J48** *aux propriétés du* **Generator** *après avoir rendu celui-ci actif (enabled).*
Nous allons utiliser le **RandomSplitResultProducer** *(apprentissage et test aléatoires) comme générateur de résultats (***Result generator***) avec 66% des données utilisés pour l'apprentissage et 34% utilisés pour les tests sur le dataset* **Iris**.
Une fois l'expérience entièrement configurée, lancez celle-ci.

Exercice 1.3 *Pour analyser les résultats, sélectionnez l'onglet* **Analyse** *en haut de la fenêtre de l'*__Experiment Environment__.
(Notez que les résultats doivent être sous le format arff. Vous pouvez visualiser le fichier d'output généré avec un éditeur de texte standard.)
Cliquez sur **Experiment** *pour analyser les résultats de l'expérience courante. Le nombre de lignes de résultat disponibles ("Got 30 results") est affiché dans le panneau* **Source**.
Cette expérience est composée de 10 runs, pour 3 classifieurs, pour 1 dataset, soit un total de 30 lignes de résultats (cf nombre de lignes dans les tables excel lors du tp précédent).

Exercice 1.4 *En appuyant sur le bouton* **Select Base**, *sélectionnez* **ZeroR** *comme étant le classifieur de base. Tous les classifieurs seront comparés à ce classifieur de base.*
Sélectionnez l'attribut $Percent_{correct}$ *du champ* **Comparison** *et cliquez sur* **Perform test** *pour générer une comparaison des 3 classifieurs.*

Comprendre:
Il existe une colonne pour chacun des classifieurs utilisés dans l'expérience, et une ligne pour chacun des datasets utilisés. Le pourcentage de "correction" pour chaque classifieur est affiché pour chaque ligne dataset: e.g. X% pour ZeroR, Y% pour OneR, et Z% pour j48.J48.
L'annotation "v" ou "*" indique qu'un résultat spécifique est statistiquement meilleur (v) ou pire (*) que le classifieur de base (dans le cas présent, ZeroR) avec un niveau de signification précisé (ici 0.05).
Les résultats de OneR et j48.J48 devraient normalement être nettement meilleurs que la base de référence établi par ZeroR.
En bas de chaque colonne (sauf pour la première) on trouve un compteur (xx/ yy/ zz) du nombre de lignes pour lequel le classifieur a été meilleur que (xx), le même que (yy), ou pire que (zz) le classifieur de base sur le dataset utilisé dans le run.
Dans l'exemple, il n'y a qu'un seul dataset et OneR a été 1 fois meilleur que ZeroR et jamais équivalent ou pire à ZeroR (1/0/0);
j48.J48 est également meilleur que ZeroR sur le dataset.
La valeur "(10)" au debut des lignes "iris" précise le nombre de runs de l'expérience.
L'écart type de l'attribut évalué peut être calculé en sélectionnant la case **std.deviations**.
En sélectionnant **Number-correct** en champ de comparaison et en cliquant sur **Perform test**, on génère la moyenne du nombre de correct sur l'ensemble des tests (sur un maximum de 51 cas de test, 34% des 150 cas dans le dataset Iris).

1.1 Rang de Test

Exercice 1.5 *Sélectionnez* **Ranking** *grâce au bouton* **Select base**.

Ceci fournit le nombre de classifieurs que chaque classifieur "surpasse" ou inversement par lesquels il est surpassé.
Le rang de test classe les classifieurs en fonction de leur nombre total de "victoires" ($>$) et de "défaites" ($<$) contre les autres classifieurs. La premiere colonne ($> - <$) fournit la différence entre le nombre de "victoires" et le nombre de "défaites".

1.2 Sauvegarder les résultats

Les informations affichées dans le panneau **Test output** (à droite de la fenêtre) sont controlées par l'entrée actuellement sélectionnée dans le panneau **Result list** (en bas à gauche de la fenêtre).

En cliquant sur l'une des entrées, les résultats correspondant à cette entrée sont affichés. Les résultats affichés dans le panneau **Test output** peuvent être sauvegardés dans un fichier en cliquant sur **Save output**.

Un seul jeu de résultats peut être sauvegardé à la fois, mais Weka permet à l'utilisateur de sauvegarder tous les résultats dans un même dataset en les sauvegardant un à la fois et en utilisant l'option *Append* au lieu de *Overwrite* pour les sauvegarder.
Le choix de cette option vous est offert via une fenêtre de dialogue apparaissant en milieu d'écran, **après** avoir sélectionné le fichier de sauvegarde.

1.3 Exercices avec le Weka Analyser

Exercice 1.6 *Définissez une experience pour comparer 3 classifieurs (IBK, J48 et OneR) par rapport au* $Percent_{correct}$*, sur les datasets Iris et Soybean.*
Souvenez-vous d'utiliser le **InstancesResultListener** *et de sauvegarder les output dans un format arff (disons Experiment5.arff).*
Une fois les expériences effectuées, utilisez les utilitaires de **Experiment Environement** *dans la partie* **Analyse**.
Pouvez-vous faire des commentaires de significations statistiques?
Vérifiez votre réponse en sélectionnant **Summary** *comme base de Test et en appuyant sur* **Perform test**.
Ceci vous fournit une matrice résumant l'analyse de l'expérience.

Comprendre:
Par exemple, une ligne (- 1 1) indique que les classifieurs liés aux colonnes "b" et "c" sont meilleurs que celui lié à la ligne "a".
Une entrée à 0 indique qu'il n'y a pas de différence significative entre le classifieur lié à la ligne et celui lié à la colonne.

Exercice 1.7 Y-a-t-il des différences significatives par rapport au temps d'apprentissage et de test entre les classifieurs testés?
Comment trouver cette information?
Indice: Modifier la sélection du "comprarison field".