Fouille de Données - TP1 : Weka, règles d'association, clustering

M2 – TIW2 – 2013-2014

Cette séance pour but de prendre en main **weka***, une plateforme d'algorithmes de data mining écrite en java que nous réutiliserons ainsi que d'expérimenter l'algorithme APriori de génération de règles d'association et les algorithmes de clustering vus en cours. Vous pouvez vous mettre en (bi|tri)nôme. Un compte-rendu sous la forme d'un document .pdf (.doc refusé ainsi que les sources Java sont à rendre avant le 11/11/2013, 23 :59. Les comptes-rendus doivent être envoyé à marc.plantevit@univ-lyon.fr avec comme objet "[M2TIW] CR TP1". N'oubliez d'indiquer les noms des membres du (mo|bi|tri)nôme dans le corps du message et le compte-rendu.*

1 Présentation et installation de Weka

Weka est un ensemble de classes et d'algorithmes en Java implémentant les principaux algorithmes de data mining. Il est disponible gratuitement à l'adresse `www.cs.waikato.ac.nz/ml/weka`, dans des versions pour Unix et Windows. Ce logiciel est développé en parallèle avec un livre : Data Mining par I. Witten et E. Frank (éditions Morgan Kaufmann). Weka peut s'utiliser de plusieurs façons :

- Par l'intermédiaire d'une interface utilisateur : c'est la méthode utilisée dans ce TP.
- Sur la ligne de commande.
- Par l'utilisation des classes fournies à l'intérieur de programmes Java : toutes les classes sont documentées dans les règles de l'art. Nous y reviendrons sans doute dans un prochain TP.

1. Téléchargez Weka et installez le.
2. Téléchargez l'excellente présentation d'Eibe Frank à `http://liris.cnrs.fr/marc.plantevit/ENS/TP/weka.ppt` et parcourez là (un tutorial sur Weka est également disponible : `http://liris.cnrs.fr/marc.plantevit/ENS/TP/Tutorial.pdf`).

2 Premiers pas

Weka est maintenant installé sur votre compte. Après l'avoir lancé, vous obtenez la fenêtre intitulée Weka GUI Chooser : choisissez l'Explorer. La nouvelle fenêtre qui s'ouvre alors (Weka Knowledge Explorer) présente six onglets :

Preprocess : pour choisir un fichier, inspecter et préparer les données.

Classify : pour choisir, appliquer et tester différents algorithmes de classification : là, il s'agit d'algorithmes de classification supervisée.

Cluster : pour choisir, appliquer et tester les algorithmes de segmentation.

Associate : pour appliquer l'algorithme de génération de règles d'association.

Select Attributes : pour choisir les attributs les plus prometteurs.

Visualize : pour afficher (en deux dimensions) certains attributs en fonctions d'autres.

3 Les données

Les données sont sous un format ARFF -pour Attribute-Relation File Format-. Des exemples de données sont disponibles une fois weka installé [1]. Ouvrez dans un éditeur un de ces fichiers d'exemples et regardez son format. Il est simple et il est facile de convertir des données-par exemple issues d'un tableur- en ARFF. (Il y a même un convertisseur inclus dans Weka du format csv vers le format arff).

Dans l'onglet Preprocess, cliquez sur Open File et ouvrez par exemple le fichier iris.arff : il contient la description de 150 spécimens d'iris de trois sortes différentes. Chaque description est composée de quatre attributs numériques

1. Au cas où, `http://liris.cnrs.fr/marc.plantevit/ENS/TP/data/`

(dimensions des sépales et des pétales), et d'un cinquième attribut qui est la classe de cet exemple (i.e. la sorte d'iris à laquelle il appartient). Pour chacun des attributs, vous pouvez obtenir, en cliquant dessus dans la sous-fenêtre Attributes, des statistiques basiques sur la répartition des valeurs pour cet attribut (sous-fenêtre Selected Attribute). On peut appliquer différents filtres aux données ; nous y reviendrons tout à l'heure.

4 Visualisation des données

Pour une première approche des données, passez dans la fenêtre Visualize. Vous y voyez un ensemble de 25 graphiques (que vous pouvez ouvrir en cliquant dessus), qui représentent chacun une vue sur l'ensemble d'exemples selon deux dimensions possibles, la couleur des points étant leur classe. Sur le graphique, chaque point représente un exemple : on peut obtenir le descriptif de cet exemple en cliquant dessus. La couleur d'un point correspond à sa classe (détaillé dans la sous-fenêtre Class colour). Au départ, le graphique n'est pas très utile, car les axes représentent le numéro de l'exemple.

1. Changez les axes pour mettre la largeur des pétales en abscisse, et la longueur des sépales en ordonnées.
2. Proposez un ensemble de deux règles simples permettant de classer les exemples selon leur genre : quelle erreur commettrez-vous ? Les petits rectangles sur la droite de la fenêtre représentent la distribution des exemples, pour l'attribut correspondant, par rapport à l'attribut (ou la classe) codé par la couleur. En cliquant du bouton gauche sur un de ces rectangles, vous le choisissez comme axe des X, le bouton droit le met sur l'axe des Y.
3. En mettant la classe sur l'axe des X, quels sont à votre avis les attributs qui, pris seuls, permettent le mieux de discriminer les exemples ? Si les points sont trop serrés, le potentiomètre Jitter, qui affiche les points "à peu près" à leur place, vous permet de les visualiser un peu plus séparément : cela peut être utile si beaucoup de points se retrouvent au même endroit du plan.

5 Un premier exemple de règle d'association

1. Lancez Weka, puis l'Explorer. Choisissez le fichier weather.nominal.arff : c'est l'exemple standard du golf (ou du tennis. . .), où tous les attributs ont été discrétisés. Les algorithmes de recherche de règles d'association se trouvent sous l'onglet Associate.
2. Choisissez l'algorithme **Apriori**.
3. Vérifiez que tout fonctionne en lançant l'algorithme sans modifier les paramètres du programme.
4. Quelles sont les informations retournées par l'algorithme ?

5.1 Modification des paramètres

En cliquant du bouton droit dans la fenêtre en face du bouton Choose, on a accès aux paramètres de l'algorithme. Le bouton More détaille chacune de ces options.

delta : fait décroître le support minimal de ce facteur, jusqu'à ce que soit le nombre de règles demandées a été trouvé, soit on a atteint la valeur minimale du support **lowerBoundMinSupport**

lowerBoundMinSupport : valeur minimale du support ($minsup$ en cours). Le support part d'une valeur initiale, et décroît conformément à delta.

metricType : la mesure qui permet de classer les règles. Supposons que L désigne la partie gauche de la règle et R la partie droite. Il y en a quatre (L désigne la partie gauche de la règle et R la partie droite) :
- Confidence : la confiance.
- Lift : l'amélioration.
- Leverage : proportion d'exemples concernés par les parties gauche et droite de la règle, en plus de ce qui seraient couverts, si les deux parties de la règles étaient indépendantes :
- Conviction : similaire à l'amélioration, mais on s'intéresse aux exemples où la partie droite de la règle n'est pas respectée. Le rapport est inversé.

minMetric : la valeur minimale de la mesure en dessous de laquelle on ne recherchera plus de règle.

numRules : Le nombre de règles que l'algorithme doit produire.

removeAllMissingCols : enlève les colonnes dont toutes le valeurs sont manquantes.

significanceLevel : test statistique

upperBoundMinSupport : valeur initiale du support.

1. Sur le fichier weather.nominal.arff, comparer les règles produites selon la mesure choisie.

6 Un deuxième exemple de règles d'association

Le fichier bank-data.cvs contient des données extraites d'un recensement de la population américaine. Le but de ces données est initialement de prédire si quelqu'un gagne plus de 50.000 dollars par an. On va d'abord transformer un peu les données :

6.1 Transformation des données

Récupérer le fichier bank-data.csv [2] Revenez à la fenêtre Preprocess.

1. Tout d'abord ouvrez le fichier bank-data.csv : il vous sera proposé d'utiliser un convertisseur : dites-oui ! Weka met à votre disposition des filtres permettant soit de choisir de garder (ou d'écarter) certains exemples, soit de modifier, supprimer, ajouter des attributs. La sous-fenêtre Filters vous permet de manipuler les filtres. Le fonctionnement général est toujours le même :
   - Vous choisissez un ensemble de filtres, chaque filtre, avec ces options, étant choisi dans le menu déroulant du haut de la sous-fenêtre, puis ajouté à la liste des filtres par la commande Add.
   - On applique les filtres avec la commande Apply Filters.
   - On peut alors remplacer le fichier précedemment chargé par les données transformées, à l'aide du bouton Replace.
   - Ce fichier devient alors le fichier de travail.
   - Le bouton Save sauvegarde ces données transformées dans un fichier.
2. Pouvez vous lancer l'algorithme Apriori ? Pourquoi ?

6.2 Sélection des attributs

Les données comportent souvent des attributs inutiles : numéro de dossier, nom,date de saisie . . . . Il est possible d'en supprimer 'à la main', à condition de connaître le domaine. On peut aussi lancer un algorithme de data mining, et regarder les attributs qui ont été utilisés : soient ceux-ci sont pertinents, et il est important de les garder, soient ils sont tellement liés à la classe qu'à eux seuls ils emportent la décision (pensez à un attribut qui serait la copie de la classe). Weka a automatisé cette recherche des attributs pertinents dans le filtre AttributeSelectionFilter, qui permet de définir les attributs les plus pertinents selon plusieurs méthodes de recherche (search), en utilisant plusieurs mesures possibles de la pertinence d'un attribut (eval).

1. Ici l'attribut id est une quantité qu'on peut ignorer pour la fouille : supprimez le !

6.3 Discrétisation

Certains algorithmes ont besoin d'attributs discrets pour fonctionner, d'autres n'acceptent que des attributs continus (réseaux de neurones, plus proches voisins). D'autres encore acceptent indifféremment des attributs des deux types. Weka dispose de filtres pour discrétiser des valeurs continues. Le filtre DiscretizeFilter permet de rendre discret un attribut continu et ceci de plusieurs façons :

- En partageant l'intervalle des valeurs possibles de l'attribut en intervalles de taille égale.
- En le partageant en intervalles contenant le même nombre d'éléments.
- En fixant manuellement le nombre d'intervalles (bins).
- En laissant le programme trouver le nombre idéal de sous intervalles.

Ici il y a plusieurs attributs numériques : "children", "income", "age".

1. Discrétiser age et income en utilisant le filtre Weka et en forçant le nombre d'intervalles à 3. Sauver le fichier transformé par exemple dans bank1.arff.
2. L'attibut children est numérique mais ne prend que 4 valeurs : 0,1,2,3 ; pour le discrétiser, on peut soit utiliser le filtre, soit le faire à la main dans le fichier arff.

Remarque : si vous éditez directement le fichier, vous pouvez en profiter pour rendre les données plus lisibles, par exemple en traduisant le nom des attributs, en donnant des noms aux intervalles obtenus par la discrétisation...

6.4 APriori

Sauvez dans bankd.arff le résultat de vos transformations : c'est le fichier qui va servir pour la génération des règles d'association.

2. http ://archive.ics.uci.edu/ml/

1. Appliquez l'algorithme Apriori et tentez d'interpréter les règles produites. Jouez sur les paramètres. Comment se comporte le temps d'exécution en fonction des paramètres ? Quels sont les paramètres les plus "critiques" ?
2. Utilisez l'algorithme Tertius. Que constatez-vous sur la forme des règles ?

6.5 Classification avec Apriori

Reprenez le fichier iris.arff ; discrétisez les attributs continus, et appliquez l'algorithme Apriori. Examinez les règles produites avec comme conclusion uniquement la classe : correspondent-elles à votre intuition ? Comparez avec ce qui est produit par un algorithme de classification, en choisissant par exemple dans trees, J48 qui construit un arbre de décision.

7 Promotions de Noël et épicerie de nuit

L'épicerie de nuit de la rue *"remplacez par votre rue favorite"* a décidé à l'approche des fêtes de fin d'année de lancer une vaste opération de promotion. Son patron, fervent adepte des nouvelles technologies et de la fouille de données (ça arrive), vous demande d'utiliser les règles d'associations pour trouver des règles intéressantes pour ses futures promotions. Il va donc réutiliser le bilan d'achats de l'année dernière à la même date :

Achats	Produit 1	Produit 2	Produit 3	Produit 4	Produit 5
Mme Michou	X			X	X
Tonton Gérard	X	X			X
Mme Guénolet					X
Mr Robert			X	X	X
Mr Sar	X	X	X	X	X
Mr causy	X				X
Mme mimi	X			X	X
Mme Fillon		X	X		

TABLE 1 – Table d'achats de l'année 2006-2007

1. Générer un fichier ARFF contenant les données du bilan d'achat
2. Extraire les règles d'associations avec un support de 0.5 puis de 0.1
3. Que pouvez-vous conseiller comme promotion au patron ?

8 Mise en œuvre

Rendez vous sur le site `http://archive.ics.uci.edu/ml/`, choisissez un jeu de données et importer le dans weka afin de le visualiser et extraire des règles d'association.

9 API Weka

Il est possible d'utiliser directement les algorithmes sur des jeux de données en les appelant directement à partir de votre propre code Java.

1. Etudiez l'API de Weka, notamment pour les règles d'Association, puis dans un programme Java, automatisez directement ce que vous avez fait via l'interface graphique en appelant directement les algorithmes nécessaires.
2. Utilisez le code précédent pour étudier le temps d'exécution de l'algorithme Apriori en fonction du seuil de support et du seuil de confiance (vous pouvez générer des graphes à l'aide de gnuplot).

Clustering – Introduction

Nous allons utiliser le paquetage *weka.clusterers* pour l'analyse de données qui ne contiennent pas d'attribut de classe. Ainsi, puisque presque tous les ensembles de données que nous utilisons possèdent un attribut de classe, nous allons ignorer ces attributs (sauf pour les phases d'évaluation). Nous allons utiliser le **Weka Knowledge Explorer**.

Remarque : Si vous souhaitez voir les commandes spécifiques des algorithmes, vous pouvez utiliser le simple CLI (command line options) : `java weka.clusterers.Cobweb -h` pour le clustering conceptuel. Ou encore : `java weka.clusterers.SimpleKMeans -h` pour le k-means clustering.

10 Premiers contacts

Les classes qui implémentent une méthode de clustering dans l'outil Weka, sont regroupées dans le package weka.clusterers. La classe weka.clusterers.Clusterer définit la structure générale commune à toutes les méthodes de clustering. A partir de l'onglet Cluster on peut observer quatre algorithmes implémentés : **SimpleKMeans**, **EM** (expectation-maximization), **CobWeb** et **FarthestFirst**.

Weka affiche le nombre d'exemples assignés à chaque cluster. Weka permet de tester la qualité du modèle sur un jeu de test. C'est la mesure de vraisemblance (log-likelihood) qui est utilisée. Plus la mesure est grande, mieux le modèle caractérise les données. Le test peut être effectué par validation croisée.

La boîte Cluster mode permet de choisir la méthode d'évaluation du modèle extrait.

Use training set : effectue et teste le clustering sur le même jeu de données ;

Supplied test set : teste le clustering sur un jeu de données à spécifier ;

Percentage split : effectue le clustering sur le pourcentage indiqué du jeu de données et teste sur le pourcentage restant ;

Classes to clusters evaluation : teste le clustering relativement à une classe

Pour cet exercice, on considérera les données du fichier **vote.arff**. Ce jeu de données décrit le résultat des votes de chaque représentant au Congrès des Etats-Unis sur les 9 questions clés identifiés par le Congressional Quarterly Almanac.

10.1 Analyse exploratoire

Effectuez une première analyse du jeu de données. Combien d'instances ? Combien d'attributs ? Quels types ? etc.

10.2 K-Means

Effectuez un clustering du jeu de données en utilisant l'algorithme **SimpleKMeans** et en conservant les paramètres par défaut.

Le résultat du clustering est donné avec une instance par cluster représentant le centroïde du cluster.

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 1510.0
Missing values globally replaced with mean/mode

Cluster centroids:
                                              Cluster#
Attribute                               Full Data          0          1
                                          (435)      (214)      (221)
=============================================================================
handicapped-infants                            n          n          y
water-project-cost-sharing                     y          y          n
adoption-of-the-budget-resolution              y          n          y
physician-fee-freeze                           n          y          n
el-salvador-aid                                y          y          n
religious-groups-in-schools                    y          y          n
anti-satellite-test-ban                        y          n          y
aid-to-nicaraguan-contras                      y          n          y
mx-missile                                     y          n          y
immigration                                    y          y          y
synfuels-corporation-cutback                   n          n          n
education-spending                             n          y          n
superfund-right-to-sue                         y          y          n
crime                                          y          y          n
duty-free-exports                              n          n          y
export-administration-act-south-africa         y          y          y
Class                                   democrat republican   democrat


Clustered Instances

0      214 ( 49%)
1      221 ( 51%)
```

Visualisation :

A partir de la boîte **Result List** (bouton droit, **Visualize cluster assignement**), visualisez la répartition des exemples dans chaque cluster.

Évaluation relativement à une classe

L'option d'évaluation **Classes to clusters evaluation** permet d'assigner une classe à un cluster pendant la phase de test. La classe assignée est la plus fréquente dans le cluster ; une erreur de classement (taux de mal classés) est calculée ainsi que la matrice de confusion. Dans ce cas, l'algorithme ne prend pas en compte la valeur de cet attribut dans le calcul de distance.

- Effectuez un clustering du jeu de données en utilisant la méthode implémentée par SimpleKMeans en conservant les paramètres par défaut avec l'option Classes to clusters evaluation et l'attribut de classe class
- Recommencez en modifiant l'attribut de classe et observer les taux d'erreur
- Conservez les meilleurs résultats et effacer les autres de la liste des résultats. Quel est l'attribut de classe pour lequel l'erreur est la plus faible ?
- Visualisez : A partir de la boîte Result List (bouton droit, Visualize cluster assignement), visualiser la répartition des exemples dans chaque cluster. Les croix représentent les instances classées dans le "bon" cluster et les carrés représentent les instances classées dans le "mauvais" cluster.
- Exportez le résultat de votre clustering (le meilleur) et caractérisez les clusters via des règles d'association.

Comparaison de modèles

- Lancez l'algorithme SimpleKMeans plusieurs fois avec les valeurs 20, 50, 100, 1000 pour le paramètre random seed avec l'option **Classes to clusters evaluation** et l'attribut de classe **class** et en fixant le nombre de clusters à 2.
- Quel est le meilleur résultat selon le taux d'erreur et la taille de clusters.
- Conservez le meilleur résultat

10.3 EM

La méthode EM (Expectation Maximisation) génère une description probabiliste des clusters en terme de moyenne et écart-type pour les attributs numériques et en terme de nombre pour les attributs nominaux. Chaque cluster est décrit par sa probabilité a priori et une distribution de probabilité pour chaque attribut. Pour un attribut nominal, est affiché le nombre d'exemples et pour un attribut numérique est affiché les caractéristiques de sa distribution normale. L'option d'évaluation **Classes to clusters evaluation** affiche aussi le **log-likelihood**, (ou log-vraissemblance) assigne une classe au cluster, calcule l'erreur et la matrice de confusion.

- Effectuez un clustering du jeu de données en utilisant la méthode EM avec les paramètres par défaut. Combien de classes sont découvertes ?

Évaluation relativement à une classe

- Effectuez un clustering du jeu de données en utilisant la méthode EM en fixant le nombre de clusters à 2 et avec l'option **Classes to clusters evaluation** et l'attribut de classe **class**.
- Effectuez un clustering du jeu de données en utilisant la méthode EM en fixant le nombre de clusters à 2 et avec l'option **Classes to clusters evaluation** et l'attribut de classe **el-salvador-aid**.
- Comparez les résultats à ceux obtenus à la question 10.2.

10.4 Clustering hiérarchiques avec Cobweb

La méthode Cobweb réalise un clustering hiérarchique où les clusters sont décrits de manière probabiliste. L'algorithme possède deux options importantes : **Cutoff** (par défaut=0.002) et **Acuity** (par défaut=1.0). ? ?

- Lancez un clustering du jeu de données en utilisant la méthode Cobweb avec les paramètres par défaut et avec l'option Classes to clusters evaluation et l'attribut de classe class La fenêtre d'output montre le nombre de noeuds regroupés puis découpés, le nombre de clusters et la structure hiérarchique de clusters. Quel est le nombre de clusters trouvés ?

Évaluation relativement à une classe

- Fixez le paramètre Cutoff à 0.5 de manière à supprimer le découpage (split) de noeuds et donc ramener le nombre de clusters à 2 et avec l'option Classes to clusters evaluation et l'attribut de classe class
- Conservez le paramètre Cutoff à 0.5 avec l'option Classes to clusters evaluation et l'attribut de classe el-salvador-aid
- Comparez les résultats obtenus avec les trois méthodes de clustering.

11 Jeux de données de base pour comparer les techniques de clustering

11.1 Jeu de données « ligne-carré »

En utilisant le jeu de données à 2 attributs x et y pour la représentation des clusters lignes-carrés (lignec1.arff et lignec2.arff),

- Testez l'algorithme des K-moyennes et celui l'Expectation-Maximization.
- Permettent-ils de retrouver les groupes identifiables graphiquement ?
- Visualisez les assignements aux clusters.

11.2 Jeu de données par distribution sur chaque attribut

En utilisant les jeux de données pour la distribution (em2.arff et em3.arff).

- Testez l'algorithme des K-moyennes et l'Expectation maximization.
- Permettent-ils de retrouver les groupes identifiables graphiquement ?
- Visualisez les assignements aux clusters.

11.3 Données Titanic

- Filtrez les données "titanic.arff" ("titanic.txt") pour enlever la prédiction (SURVIVED) des données d'apprentissage.
- Les algorithmes de clustering des K-moyennes et l'Expectation maximization permettent-ils de découper les données en un groupe de survivants et un groupe de non-survivants ?
- Visualisez les assignements aux clusters.

11.4 Iris

- Testez les algorithmes des K-moyennes et l'Expectation maximization, en faisant varier les paramètres sur le jeu de données Iris.
- Ignorez des attributs et tester l'influence sur le résultat.
- Visualisez les assignements aux clusters.

11.5 Bilan

- Testez si possible d'autres jeux de données (de préférence les plus grands i.d. soybean, labour, etc.) et analysez les différents clusters fournis avec SimpleKMeans, EM et Cobweb.
- Finalement, listez les principaux avantages et désavantages de ces algorithmes de clustering utilisés durant la séance.
- Evaluer la qualité d'un clustering est toujours difficile lorsque l'on compare différentes exécutions. D'après ce que vous avez constaté durant cette séance, quels critères pourriez-vous employer pour la qualite ? des clusters.

12 Application

Cet exercice porte sur les jeux de données Clients (Clients.csv) et Immatriculations (Immatriculations.csv) Voici quelques informations sur les attributs

Pour Clients.csv : ?

- taux : exprime en euros la capacité d'endettement du client (correspond à environ 30% de son salaire)
- 2eme voiture : valeur booléenne indiquant si le client possédait déjà un véhicule principal

Pour Immatriculations.csv :

- puissance : exprimée en chevaux
- longueur : quatre catégories ont été déterminées « courte », « moyenne », « longue » et « très longue »
- prix : en euros

On demande de lancer différentes méthodes de clustering sur ces deux jeux de données et de présenter les solutions les plus pertinentes extraites (on pourra considérer le jeu de données Clients_1.csv dans lequel l'attribut immatriculation a été supprimé). Il s'agira de sélectionner différents attributs des jeux de données initiaux et de lancer différents algorithmes de clustering en faisant varier la valeurs de leurs paramètres

13 Utilisation de weka sans l'interface

Dans cette partie, on va utiliser les classes Java de Weka (voir la doc en ligne).

- Vous avez dû constater que l'algorithme K-Means est sensible à l'initialisation. Écrivez un programme Java qui prend en argument un fichier de données et applique l'algorithme avec différentes graines, et retourne pour chaque valeur l'erreur "SquaredError". Vous pouvez retournez la meilleure segmentation.
- Un autre inconvénient de l'algorithme K-Means est de devoir fixer le nombre de clusters. Écrivez un programme qui prend en argument un fichier de données et qui applique l'algorithme avec différents nombres de clusters. Pour chaque valeur, vous pourrez appliquer l'algorithme avec différentes graines et prendre l'erreur moyenne. Affichez les différentes erreurs moyennes pour chaque valeur de nombre de clusters (k). Testez sur les exemples précédents. Que constatez vous ? Cette approche peut-elle vous aider à trouver le nombre « idéal » de clusters pour le jeu de données considéré ?

Remarque : pour interpréter plus facilement les résultats, on peut visualiser le graphe de la fonction qui associe l'erreur moyenne au nombre de clusters. Pour cela, vous pouvez utiliser **gnuplot**. *Il suffit de créer un fichier contenant les valeurs (une ligne par point de la forme x, y et sous gnuplot de l'afficher avec la commande* `plot nomfichier with lines`. *Pour ceux qui sont sous Windows, il vous reste les outils de bureautique habituels.*

14 Knime

Lors de la prochaine séance, nous utiliserons la plateforme KNIME qui subsume Weka, R et possède beaucoup plus de fonctionnalités que Weka. L'une de ses principales caractéristiques est la définition de workflow.

- **Veillez à télécharger et installer KNIME (`https://www.knime.org/`) avant la prochaine séance.**