Proceedings of the 4th National Conference; INDIACom-2010
Computing For Nation Development, February 25 – 26, 2010
Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi

K-MEANS CLUSTERING USING WEKA INTERFACE

1.Sapna Jain
PhD Fellow
Department of Computer Science
Jamia Hamdard University
New Delhi-110062,India
hellosap@sify.com

2.M Afshar Aalam
Professor,Head
Department of Computer Science
Jamia Hamdard University
New Delhi-110062,India
mailtoafshar@rediffmail.com

3. M. N Doja
Professor
Faculty of Engg & Technology
Jamia Millia Islamia,Jamia Nagar
New Delhi – 110025(INDIA)
mndoja@gmail.com

ABSTRACT

Weka is a **landmark system in the history of the data mining and machine learning** research communities,because it is the only toolkit that has gained such widespread adoption and survived for an extended period of time (the first version of Weka was released 11 years ago). Other data mining and machine learning systems that have achieved this are individual systems, such as C4.5, not toolkits.Since Weka is freely available for download and offers many powerful features (sometimes not found in commercial data mining software), it has become one of the most widely used data mining systems. Weka also became one of the favorite vehicles for data mining research and helped to advance it by making many powerful features available to all.

This paper provides a comprehensive review of K-means clustering techniques in WEKA 3.7.More than twelve years have elapsed since the first public release of WEKA. In that time, the software has been rewritten entirely from scratch, evolved downloaded more than 1.4 million times since being placed on Source-Forge in April 2000. This paper provides an introduction to the WEKA workbench, reviews the history of the project,and, in light of the recent 3.7 stable release,k-means clustering execution in WEKA 3.7.

Key words: Weka3.7, Cluster analysis, *k*-means algorithm, Euclidean Distance.

INTRODUCTION

The *Weka* or *woodhen* (Gallirallus australis) is an endemic bird of New Zealand. The WEKA project aims to provide a comprehensive collection of machine learning algorithms and data preprocessing tools to researchers and practitioners alike. It allows users to quickly try out and compare different machine learning methods on new data sets. Its modular, extensible architecture allows sophisticated data mining processes to be built up from the wide collection of base learning algorithms and tools provided. Extending the toolkit is easy thanks to a simple API, plugin mechanisms and facilities that automate the integration of new learning algorithms with WEKA's graphical user interfaces.

The Weka Data Mining Software has been downloaded **200,000 times** since it was put on SourceForge in April 2000, and is currently downloaded at a rate of 10,000/month. The Weka mailing list has over **1100 subscribers in 50 countries**, including subscribers from many major companies.
There are **15 well-documented substantial projects** that incorporate, wrap or extend Weka, and no doubt many more that have not been reported on Sourceforge.Ian H. Witten and Eibe Frank also wrote a **very popular book "Data Mining: Practical Machine Learning Tools and Techniques"** (now in the second edition), that seamlessly integrates Weka system into teaching of data mining and machine learning. The **key features** responsible for Weka's success are:
– it provides many different algorithms for data mining and machine learning.
– is is open source and freely available.
– it is platform-independent.
– it is easily useable by people who are not data mining specialists.
– it provides flexible facilities for scripting experiments
– it has kept up-to-date, with new algorithms being added as they appear in the research literature.
In sum, the Weka team has made an outstanding contribution to the data mining field.

2 WEKA INTERFACE

. Figure1 .Weka Interface

The Weka GUI Chooser (class weka.gui.GUIChooser) provides a starting point for launching Weka's main GUI applications and supporting tools. If one prefers a MDI ("multiple document interface") appearance, then this is provided by an alternative launcher called "Main" (class weka.gui.Main).

The GUI Chooser consists of four buttons—one for each of the four major Weka applications—and four menus.The buttons can be used to start the following applications:

• Explorer : An environment for exploring data with WEKA .
• Experimenter : An environment for performing experiments and conducting statistical tests between learning schemes.
• KnowledgeFlow : This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.
• SimpleCLI : Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

3. WEKA CLUSTERER

It contains "clusterers" for finding groups of similar instances in a dataset.Some implemented schemes are: *k*-Means, EM, Cobweb, *X*-means, FarthestFirst .Clusters can be visualized and compared to "true" clusters (if given)Evaluation based on log likelihood if clustering scheme produces a probability distribution.

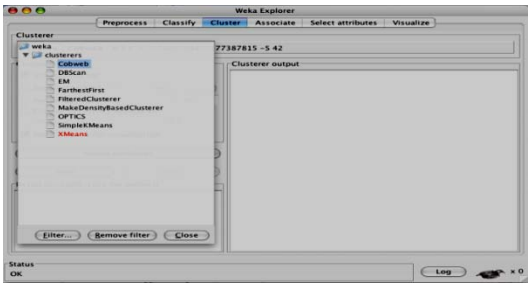

Figure 2 Weka Clusterer

3.1 CLUSTERING CONCEPT

Clustering is a division of data into groups of similar objects. Representing the data by fewer clusters necessarily loses certain fine details, but achieves simplification. It models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. From a practical perspective clustering plays an outstanding role in data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, CRM, marketing,medical diagnostics, computational biology, and many others.Clustering is widely used in gene expression data analysis. By grouping genes together based on the similarity between their gene expression proles, functionally related genes may be found. Such a grouping suggests the function of presently unknown genes.

3.2 CLUSTERING TECHNIQUES

Traditionally clustering techniques are broadly divided in hierarchical and partitioning.Hierarchical clustering is further subdivided into agglomerative and divisive. The basics of hierarchical clustering include Lance-Williams formula, idea of conceptual clustering,
now classic algorithms SLINK, COBWEB, as well as newer algorithms CURE and CHAMELEON.

While hierarchical algorithms build clusters gradually (as crystals are grown),partitioning algorithms learn clusters directly. In doing so, they either try to discover clusters by iteratively relocating points between subsets, or try to identify clusters as areas highly populated with data.

Partitioning Relocation Methods. They are further categorized into probabilistic clustering (EM framework, algorithms SNOB, AUTOCLASS, MCLUST), k-medoids methods (algorithms PAM, CLARA, CLARANS, and its extension), and k-means methods (different schemes, initialization, optimization, harmonic means, extensions).Such methods concentrate on how well points fit into their clusters and tend to build clusters of proper convex shapes.

Partitioning algorithms of the second type are surveyed in the section Density-Based Partitioning. They try to discover dense connected components of data, which are flexible in terms of their shape. Density-based connectivity is used in the algorithms DBSCAN, OPTICS, DBCLASD, while the algorithm DENCLUE exploits space density functions. These algorithms are less sensitive to outliers and can discover clusters of irregular shapes. They usually work with low-dimensional data of numerical attributes,known as spatial data. Spatial objects could include not only points, but also extended objects (algorithm GDBSCAN).

4. K-MEANS CLUSTERING TECHNIQUE

The basic step of k-means clustering is simple. In the beginning, we determine number of cluster K and we

assume the centroid or center of these clusters. We can take any random objects as the initial centroids or the first K objects can also serve as the initial centroids.
Then the K means algorithm will do the three steps below until convergence.Iterate until *stable* (= no object move group):
1. Determine the centroid coordinate
2. Determine the distance of each object to the centroids
3. Group the object based on minimum distance (find the closest centroid)

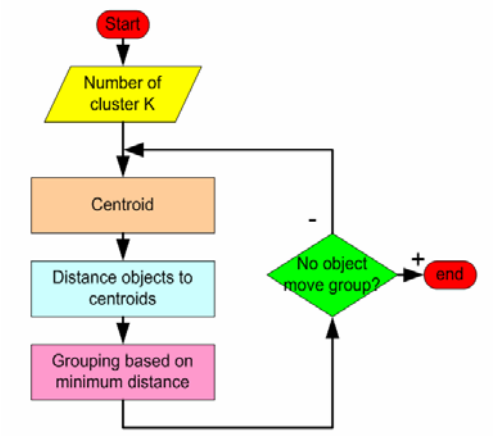

Figure 3: K-means clustering process.

4.2 DISTANCE CALCULATION

Euclidean Distance is the most common use of distance. In most cases when people said about distance , they will refer to Euclidean distance. Euclidean distance or simply 'distance' examines the *root of square differences* between coordinates of a pair of objects.

Formula

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk} \right)^2}$$

For example:

Features k

	Cost	Time	Weight	Incentive
Object A	0	3	4	5
Object B	7	6	3	-1

Point A has coordinate (0, 3, 4, 5) and point B has coordinate (7, 6, 3, -1).The Euclidean Distance between point A and B is

$$d_{BA} = \sqrt{(0-7)^2 + (3-6)^2 + (4-3)^2 + (5+1)^2}$$
$$= \sqrt{49+9+1+36} = 9.747$$

Euclidean distance is a special case of Minkowski distance with $\lambda = 2$

5 K-means in WEKA 3.7

This example illustrates the use of *k-means* clustering with WEKA The sample data set used for this example is based on the "bank data" available in comma-separated format bank-data.csv. This paper assumes that appropriate data preprocessing has been perfromed. In this case a version of the initial data set has been created in which the ID field has been removed and the "children" attribute has been converted to categorical.

The resulting data file is “bank.arff” and includes 600 instances. As an illustration of performing clustering in WEKA, we will use its implementation of the K-means algorithm to cluster the cutomers in this bank data set, and to characterize the resulting customer segments.

Figure 4 shows the main WEKA Explorer interface with the data file loaded.

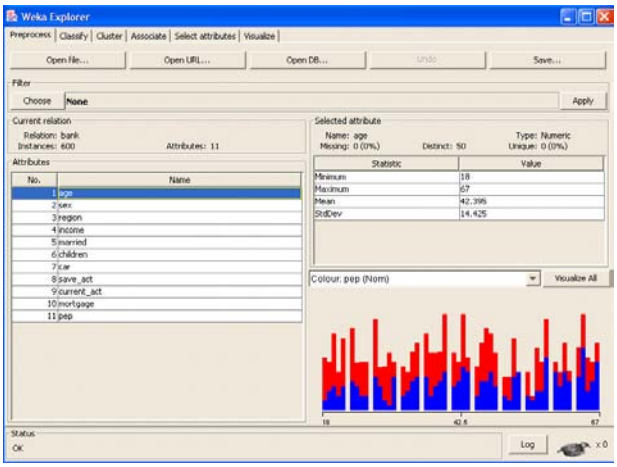

Figure 4 Data file Bank.data.csv loaded

Some implementations of K-means only allow numerical values for attributes. In that case, it may be necessary to convert the data set into the standard spreadsheet format and convert categorical attributes to binary. It may also be necessary to normalize the values of attributes that are measured on substantially different scales (e.g., "age" and "income"). While WEKA provides filters to accomplish all of these preprocessing tasks, they are **not necessary for clustering in WEKA** . This is because WEKA SimpleKMeans algorithm automatically handles a mixture of categorical and numerical attributes.

Furthermore, the algorithm automatically normalizes numerical attributes when doing distance computations. The WEKA SimpleKMeans algorithm uses Euclidean distance measure to compute distances between instances and clusters.

To perform clustering, select the "Cluster" tab in the Explorer and click on the "Choose" button. This results in a drop down list of available clustering algorithms. In this case we select "SimpleKMeans". Next, click on the text box to the right of the "Choose" button to get the pop-up window shown in Figure 5, for editing the clustering parameter.

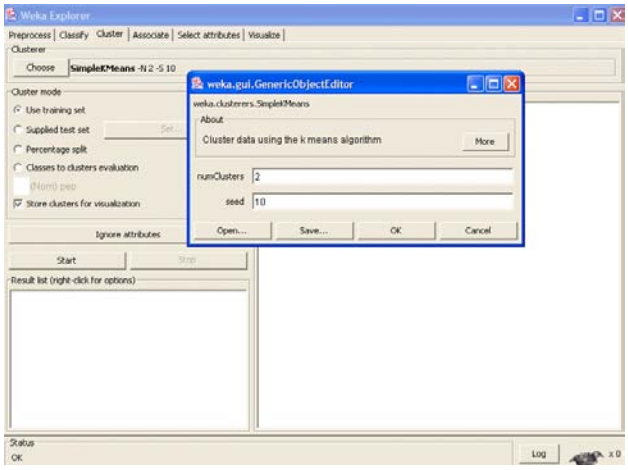

Figure 5 Selecting Clustering Parameters.

In the pop-up window we enter **6** as the number of clusters (instead of the default values of 2) and we leave the value of "seed" as is. The seed value is used in generating a random number which is, in turn, used for making the initial assignment of instances to clusters. Note that, in general, K-means is quite sensitive to how clusters are initially assigned. Thus, it is often necessary to try different values and evaluate the results.

Once the options have been specified, we can run the clustering algorithm. Here we make sure that in the "Cluster Mode" panel, the "Use training set" option is selected, and we click "Start". We can right click the result set in the "Result list" panel and view the results of clustering in a separate window. This process and the resulting window are shown in Figures 6 and Figure7.

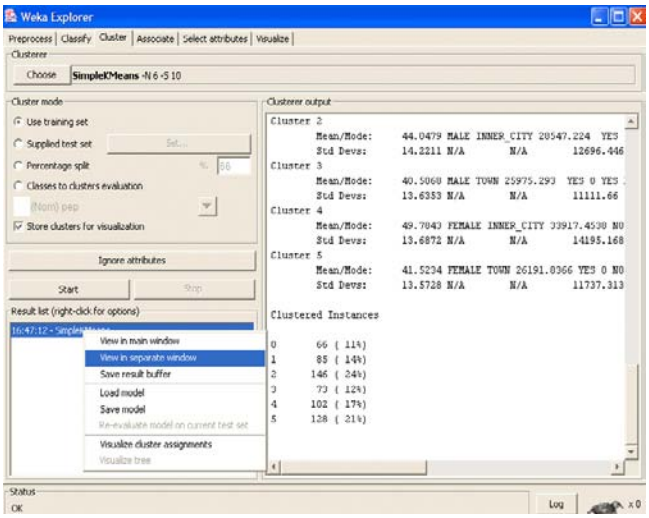

figure6 Clustering in progress.

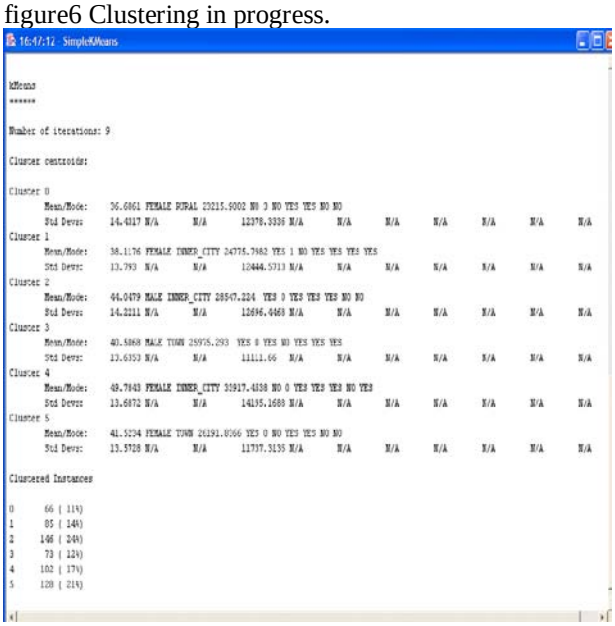

Figure 7 Clustering data ouput

The result window shows the centroid of each cluster as well as statistics on the number and percentage of instances assigned to different clusters. Cluster centroids are the mean vectors for each cluster (so, each dimension value in the centroid represents the mean value for that dimension in the cluster). Thus, centroids can be used to characterize the clusters. For example, the centroid for cluster 1 shows that this is a segment of cases representing middle aged to young (approx. 38) females living in inner city with an average income of approx. $28,500, who are married with one child, etc. Furthermore, this group have on average said YES to the PEP product.

Another way of understanding the characteristics of each cluster in through visualization. We can do this by right-clicking the result set on the left "Result list" panel and selecting "Visualize cluster assignments". This pops up the visualization window as shown in Figure

8.

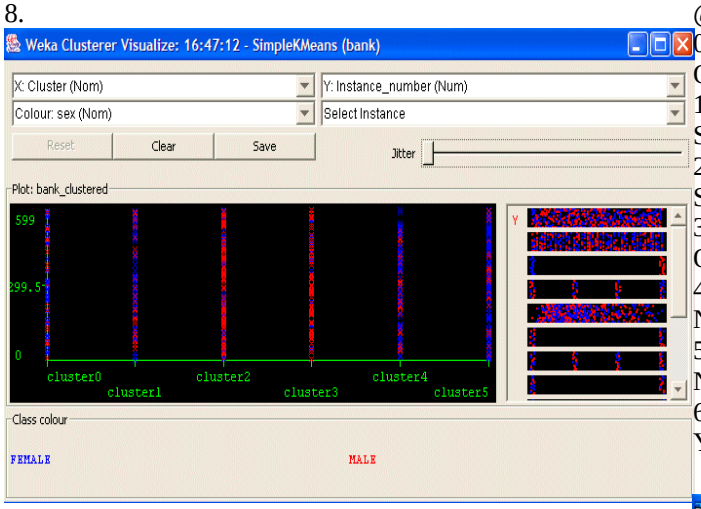

Figure 8 Visualizing Clusters .

You can choose the cluster number and any of the other attributes for each of the three different dimensions available (x-axis, y-axis, and color). Different combinations of choices will result in a visual rendering of different relationships within each cluster. In the above example, we have chosen the cluster number as the x-axis, the instance number (assigned by WEKA) as the y-axis, and the "sex" attribute as the color dimension. This will result in a visualization of the distribution of males and females in each cluster. For instance, you can note that clusters 2 and 3 are dominated by males, while clusters 4 and 5 are dominated by females. In this case, by changing the color dimension to other attributes, we can see their distribution within each of the clusters.

Finally, we may be interested in saving the resulting data set which included each instance along with its assigned cluster. To do so, we click the "Save" button in the visualization window and save the result as the file “bank-kmeans.arff “.

Bank-kmeans.arff @relation bank_clustered

@attribute Instance_number numeric
@attribute age numeric
@attribute sex {FEMALE,MALE}
@attribute region
{INNER_CITY,TOWN,RURAL,SUBURBAN}
@attribute income numeric
@attribute married {NO,YES}
@attribute children {0,1,2,3}
@attribute car {NO,YES}
@attribute save_act {NO,YES}
@attribute current_act {NO,YES}
@attribute mortgage {NO,YES}
@attribute pep {YES,NO}
@attributecluster
{cluster0,cluster1,cluster2,cluster3,cluster4,cluster5}

@data
0,48,FEMALE,INNER_CITY,17546,NO,1,NO,NO,NO,NO,YES,cluster1
1,40,MALE,TOWN,30085.1,YES,3,YES,NO,YES,YES,NO,cluster3
2,51,FEMALE,INNER_CITY,16575.4,YES,0,YES,YES,YES,NO,NO,cluster2
3,23,FEMALE,TOWN,20375.4,YES,3,NO,NO,YES,NO,NO,cluster5
4,57,FEMALE,RURAL,50576.3,YES,0,NO,YES,NO,NO,NO,cluster5
5,57,FEMALE,TOWN,37869.6,YES,2,NO,YES,YES,NO,YES,cluster5
6,22,MALE,RURAL,8877.07,NO,0,NO,NO,YES,NO,YES,cluster0

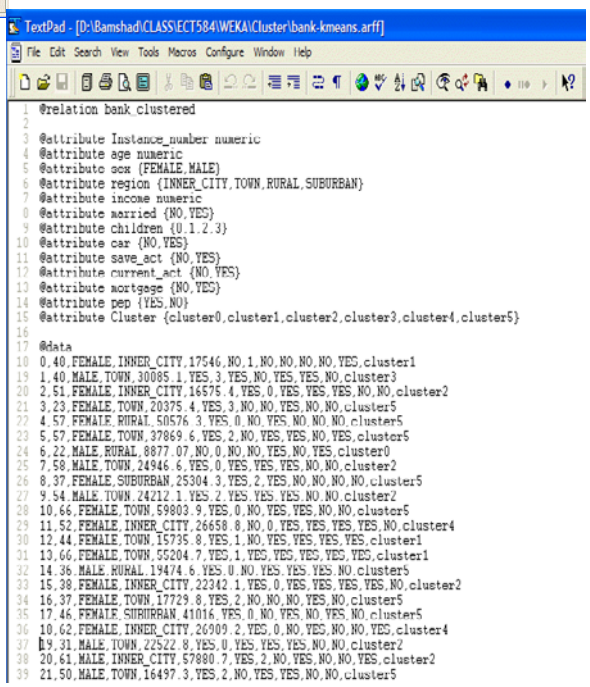

Figure 9. The top portion of this file bank.k-means.arff

Note that in addition to the "instance_number" attribute, WEKA has also added "Cluster" attribute to the original data set. In the data portion, each instance now has its assigned cluster as the last attribute value. By doing some simple manipulation to this data set, we can easily convert it to a more usable form for additional analysis or processing. For example, here we have converted this data set in a comma-separated format and sorted the result by clusters. Furthermore, we have added the ID field from the original data set (before sorting). The results of these steps can be seen in the file “bank-kmeans.csv” .

6. WEKA TEAM ACHIEVEMENT

SIGKDD Service Award is the highest service award in the field of data mining and knowledge discovery. It is is given to one individual or one group who has

performed significant service to the data mining and knowledge discovery field, including professional volunteer services in disseminating technical information to the field, education, andresearch funding. The **2005 ACM SIGKDD Service Award** is presented to **the Weka team** for their development of the freely-available Weka Data Mining Software, including the accompanying book Data Mining: Practical Machine Learning Tools and Techniques (now in second edition) and much other documentation. The Weka team includes **Ian H. Witten** and **Eibe Frank**, and the following major contributors (in alphabetical order of last names): Remco R. Bouckaert, John G. Cleary, Sally Jo Cunningham, Andrew Donkin, Dale Fletcher, Steve Garner, Mark A. Hall, Geoffrey Holmes, Matt Humphrey, Lyn Hunt, Stuart Inglis, Ashraf M. Kibriya, Richard Kirkby, Brent Martin, Bob McQueen, Craig G. Nevill-Manning, Bernhard Pfahringer, Peter Reutemann, Gabi Schmidberger, Lloyd A. Smith, Tony C. Smith, Kai Ming Ting, Leonard E. Trigg, Yong Wang, Malcolm Ware, and Xin Xu.The Weka team has put a tremendous amount of effort into continuously developing and maintaining the system **since 1994**. The development of Weka was funded by a grant from the New Zealand Government's Foundation for Research, Science and Technology.

7 CONCLUSION

WEKA's support for clustering tasks is as extensive as its support for classification and regression and it has more techniques for clustering than for association rule mining,which has up to this point been somewhat neglected. WEKA support various clustering algorithms execution in Java which gives a platform for data mining research process.Releasing WEKA as open source software and implementing it in Java has played no small part in its success.

8.FUTURE SCOPE

We are following the Linux model of releases, where an even second digit of a release number indicates a "stable" release and an odd second digit indicates a "development" release (e.g. 3.0.x is a stable release, and 3.1.x is a developmental release).

9. ACKNOWLEDGMENT

Many thanks to past and present members of the Waikato machine learning group and the external contributers for all the work they have put into WEKA.

REFERENCES

[1] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludscher,and S. Mock. Kepler: An extensible system for design and execution of scientific workflows. In In SSDBM,pages 21–23, 2004.

[2] K. Bennett and M. Embrechts. An optimization perspective on kernel partial least squares regression. In J. S. et al., editor, Advances in Learning Theory: Methods,Models and Applications, volume 190 of NATO Science Series, Series III: Computer and System Sciences,pages 227–249. IOS Press, Amsterdam, The Netherlands,2003.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J.Stone. Classification and Regression Trees. Wadsworth International Group, Belmont, California, 1984.

[4] S. Celis and D. R. Musicant. Weka-parallel: machine learning in parallel. Technical report, Carleton College,CS TR, 2002.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[6] T. G. Dietterich, R. H. Lathrop, and T. Lozano-P´erez. Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell., 89(1-2):31–71, 1997.

[7] J. Dietzsch, N. Gehlenborg, and K. Nieselt. Maydaya microarray data analysis workbench. Bioinformatics,22(8):1010–1012, 2006.

[8] L. Dong, E. Frank, and S. Kramer. Ensembles of balanced nested dichotomies for multi-class problems. In Proc 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal,pages 84–95. Springer, 2005.

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. Journal of Machine Learning. Research, 9:1871–1874, 2008.

[10] E. Frank and S. Kramer. Ensembles of nested dichotomies for multi-class problems. In Proc 21st International Conference on Machine Learning, Banff, Canada, pages 305–312. ACM Press, 2004.

[11] R. Gaizauskas, H. Cunningham, Y. Wilks, P. Rodgers,
and K. Humphreys. GATE: an environment to support research and development in natural language engineering.In In Proceedings of the 8th IEEE International Conference on Tools with Artificial Intelligence, pages 58–66, 1996.

[12] J. Gama. Functional trees. Machine Learning, 55(3):219–250, 2004.

[13] A. Genkin, D. D. Lewis, and D. Madigan. Largescale bayesian logistic regression for text categorization.Technical report, DIMACS, 2004.

[14] J. E. Gewehr, M. Szugat, and R. Zimmer. BioWeka—extending the weka framework for bioinformatics.Bioinformatics, 23(5):651–653, 2007.

[15] M. Hall and E. Frank. Combining naive Bayes and decision tables. In Proc 21st Florida Artificial Intelligence Research Society Conference, Miami, Florida. AAAI Press, 2008.

[16] K. Hornik, A. Zeileis, T. Hothorn, and C. Buchta. RWeka: An R Interface to Weka, 2009. R package version 0.3-16.