International Journal of Advanced Technology & Engineering Research (IJATER)
www.ijater.com

COMPARATIVE STUDY OF EM AND K-MEANS CLUSTERING TECHNIQUES IN WEKA INTER-FACE

Namita Bhan, M.Tech (CSE), Amity University; Prof. (Dr.) Deepti Mehrotra, Amity School of Computer Sciences

Abstract

Clustering is the fundamental task in data mining. It is a technique by which we can categorize between the similar and the dissimilar objects and group the ones together that are more likely to each other. As per the ICDM'06, k-Means holds a rank 2 and EM holds a rank 5 in the Top 10 Algorithms. This paper provides the detailed comparative study of the k-Means and the Expectation Maximization Technique using the Weka 3.6.9 Interface. Weka is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka is free software available under the GNU General Public License. The data used is the fixed broadband Internet Users for 214 Countries from year 1998-2011. Fixed broadband Internet subscribers are the number of broadband subscribers with a digital subscriber line, cable modem, or other high-speed technology.

Keywords: K-Means, EM, Cluster Analysis, Weka

Introduction

The Weka or woodhen (Gallirallus australis) is a flightless bird species of the rail family and is an endemic bird of New Zealand. As per the KDD Nuggets Survey in May 2010, Weka is one of the tools that is widely used by companies in real projects for the data analysis. The results of the Survey were based on the response of the 900 respondents. In yet another survey by Rexler Analyics, 2010, WEKA is one of the top 5 most used tools.

Clustering

Clustering is the process applied on datasets to partition the data into various meaningful subsets, called as the Clusters. The objects within each clusters share a common trait. The goal of the Cluster Analysis is descriptive and aims to discover a new set of categories. Clustering is about finding the 'similarity' and to find how similar the two objects, the distance measure is used. The objects that are within the same cluster need to be close to each other. Hence, in case of the similar objects the distance measure will be a short distance.

Distance Measure

There are different ways to define a distance Measure. Some elements may be close to each other as per one definition of the distance measure and might be far from each other as per the other definition of the distance. Some of the most commonly used distance functions are:

- Euclidean Distance / 2-Norm/ Crow distance
- Manhattan Distance / 1-Norm/ Taxicab distance

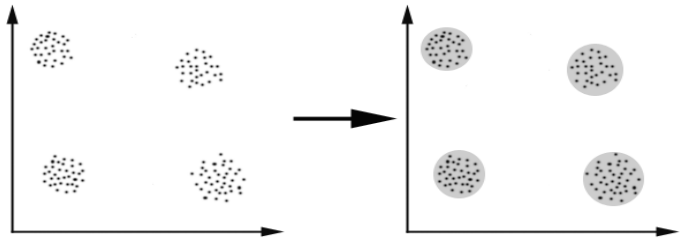

Figure 1. Formation of Cluster Groups

Introduction to Weka

Weka stands for Waikato Environment for Knowledge Analysis. Weka is the most widely tool for the data mining. It is an open source interface developed in Java and for the past 12 years, there have been various releases of the Interface. With each release, the latest algorithms for data mining are incorporated in the interface.

Main features of WEKA include:

- 49 data preprocessing tools
- 76 classification/regression algorithms
- 8 clustering algorithms
- 15 attribute/subset evaluators + 10 search algorithms for feature selection
- 3 algorithms for finding association rules
- 3 graphical user interfaces
  - ✓ "The Explorer" (exploratory data analysis)
  - ✓ "The Experimenter" (experimental environment)
  - ✓ "The Knowledge Flow" (new process model inspired interface)

Evolution of Weka

Weka was developed under a project funded by the NZ Government since 1993. The Programme goal was to build state-of-the-art facilities for developing techniques of machine learning and investigating their application in key areas of New Zealand economy.

The timelines of the various stages of Weka Development History are as follows:

- Late 1992 - funding was applied for by Ian Witten
- 1993 - development of the interface and infrastructure
  - WEKA acronym coined by Geoff Holmes
  - WEKA's file format "ARFF" was created by Andrew Donkin
    - ARFF was rumored to stand for Andrew's Ridiculous File Format
- Sometime in 1994 - first internal release of WEKA
  - TCL/TK user interface + learning algorithms written mostly in C
  - Very much beta software
    - Changes for the b1 release included (among others)
- October 1996 - first public release of WEKA (v 2.1)
- July 1997 - WEKA 2.2
  - Schemes: 1R, T2, K*, M5, M5Class, IB1-4, FOIL, PEBLS, support for C5
  - Included a facility (based on unix makefiles) for configuring and running large scale experiments
- Early 1997 - decision was made to rewrite WEKA in Java.
  - Originated from code written by Eibe Frank for his PhD
  - Originally codenamed JAWS (Java Weka System)
- May 1998 - WEKA 2.3
  - Last release of the TCL/TK-based system
- Mid 1999 - WEKA 3 (100% Java) released
  - Version to complement the Data Mining book
  - Development version (including GUI)

Weka 3.6.9

Weka aims at providing an interface where various algorithms are incorporated for the practitioners for research. Users can quickly adapt the interface due to its simplicity and ease of use.

Figure 2. WEKA Interface

The Weka toolkit has been developed in Java and the Weka GUI Chooser has four options:

• Explorer: An environment for exploring data with WEKA (the rest of this documentation deals with this application in more detail).

• Experimenter: An environment for performing experiments and conducting statistical tests between learning schemes.

• Knowledge Flow: 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.

• Simple CLI: 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.

The analysis was performed in WEKA Explorer. The Weka Interface consists of 6 tabs. When the Explorer is first started only the first tab is active; the others are grayed out. This is because it is necessary to open (and potentially pre-process) a data set before starting to explore the data.

The tabs are as follows:

• Preprocess: Choose and modify the data being acted on.
• Classify: Train and test learning schemes that classify or perform regression.
• Cluster: Learn clusters for the data.
• Associate: Learn association rules for the data.
• Select attributes: Select the most relevant attributes in the data.
• Visualize: View an interactive 2D plot of the data.

K-Means Clustering Technique

In data mining, k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space into Voronoi cells.

Given a set of observations (x1, x2, …, xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k sets ($k \leq n$) S = {S1, S2, …, Sk} so as to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2 \qquad (1)$$

where μi is the mean of points in Si.

Algorithm:

- Place K points into the space of the objects being clustered. They represent the initial group centroids.
- Assign each object to the group that has the closest centroid.
- Recalculate the positions of the K centroids.

Repeat Steps 2 & 3 until the group centroids no longer move

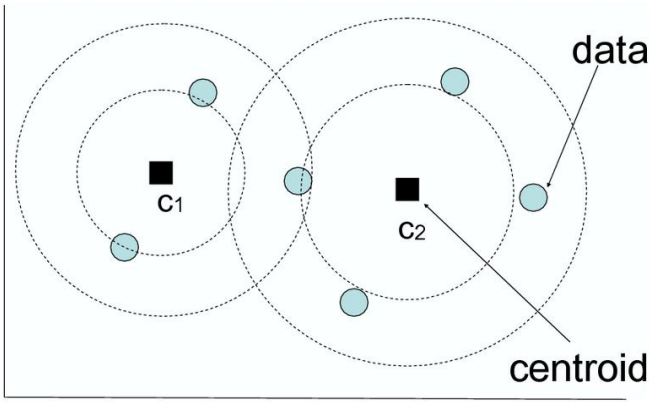

Figure 3. K-Means Clsutering

Expectation-Maximization Technique

Here, the data set is usually modeled with a fixed (to avoid overfitting) number of Gaussian distributions that are initialized randomly and whose parameters are iteratively optimized to fit better to the data set. This will converge to a local optimum, so multiple runs may produce different results. In order to obtain a hard clustering, objects are often then assigned to the Gaussian distribution they most likely belong to, for soft clustering this is not necessary.

The Expectation-Maximization (EM) algorithm is an optimization procedure which computes the Maximal-Likelihood (ML) estimate of the unknown parameter $\theta \in \Theta$ when only uncomplete (y is unknown) data $\mathcal{T}_X$ are presented. In other words, the EM algorithm maximizes the likelihood function

$$l(\theta|\mathcal{T}_X) = P(\mathcal{T}_X|\theta) = \prod_{i=1}^{l} P(x_i|\theta) = \prod_{i=1}^{l} \sum_{y \in \mathcal{Y}} P(x_i|y,\theta)P(y|\theta) . \quad (2)$$

with respect to the parameter $\theta \in \Theta$.

Experimentation

The K-Means and the EM clustering technique is performed in Weka 3.6.9. Data is first cleaned such that the missing values are replaced by the value '0'. The data used is the fixed broadband Internet Users for 214 Countries from year 1998-2011. Fixed broadband Internet subscribers are the number of broadband subscribers with a digital subscriber line, cable modem, or other high-speed technology. The preprocessing and the classification follow later.

- **Clustering using k-Means:**

The Simple k-Means technique when applied on our dataset results in the formation of 2 clusters with 210 instances lying in one cluster and the remaining instances in the other cluster.

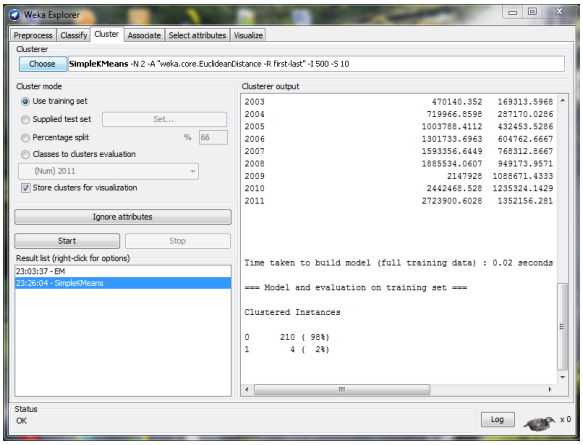

Figure 4. K-Means –Cluster Mode (Training Set)

The results of the algorithm in the tool are:

Number of iterations: 7
Within cluster sum of squared errors: 432.38667492973264
Time taken to build model (full training data): 0.02 seconds
Clustered Instances

0 210 (98%)
1 4 (2%)

In the k-Means technique only, the another variant has been

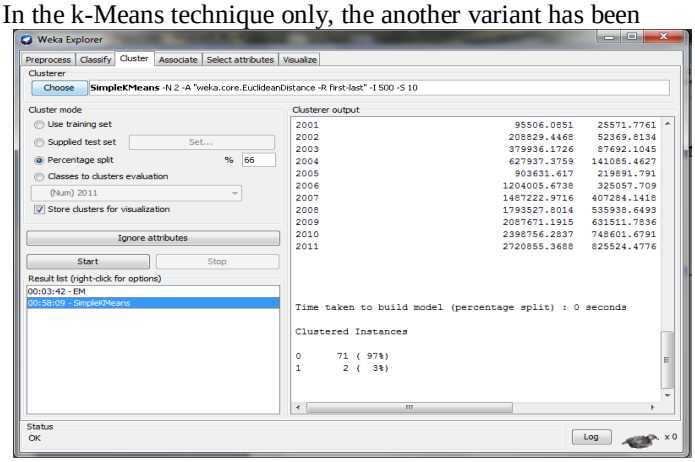

Figure 5. K-Means –Cluster Mode (Percentage Split Set)

Number of iterations: 5
Within cluster sum of squared errors: 288.7113044311968
Time taken to build model (full training data): 0.01 seconds
Clustered Instances

0 71 (97%)
1 2 (3%)

- **Clustering using EM technique:**

The clustering performed using the EM Technique in Weka was initially done using the full training set which resulted in 9 clusters and for the same dataset a percentage split at 66% resulted in 2 clusters.

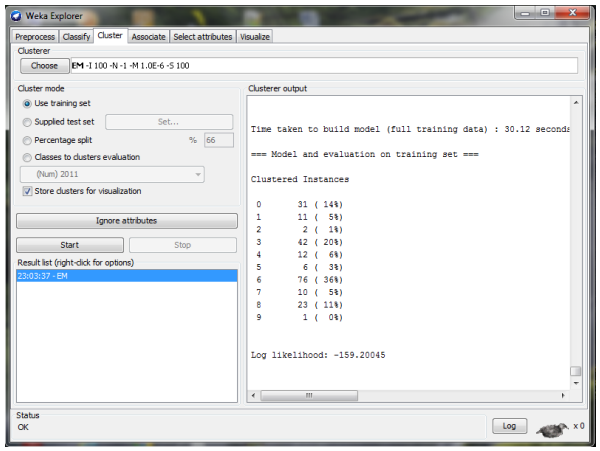

Figure 6. EM Technique –Cluster Mode (Training Set)

Time taken to build model (full training data): 30.12 seconds
Clustered Instances

0 31 (14%)
1 11 (5%)
2 2 (1%)
3 42 (20%)
4 12 (6%)
5 6 (3%)
6 76 (36%)
7 10 (5%)
8 23 (11%)
9 1 (0%)

Log likelihood: -159.20045

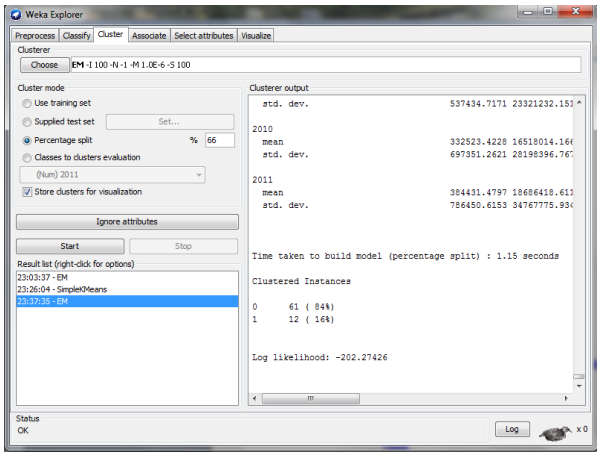

Figure 7. EM Technique–Cluster Mode (Percentage Split Set)

Time taken to build model (percentage split): 1.15 seconds Clustered Instances

0 61 (84%)
1 12 (16%)

Log likelihood: -202.27426

Results

Using Simple k-Means:

Time taken to build model (full training data): 0.02 seconds
Time taken to build model (percentage split): 0.01 seconds

Using EM Technique:

Time taken to build model (full training data): 30.12 seconds
Time taken to build model (percentage split): 1.15 seconds

Summary and Conclusion

The K-Means algorithm is very bad at handling overlapping data points. This is because it is only able to classify a point based on its distance from the estimated means. When the data overlaps, there is no clear line that can be drawn to separate those points that are closest to one mean versus those points that are closest to another.

On the other hand, EM does much better on the overlapping data. This is because the strength of EM lies in the fact that it is able to incorporate underlying assumptions about how the data was generated. The algorithm as implemented assumes that the data was generated by exactly two Gaussians. Using this knowledge, it can make a better guess as to the likely source of the data because it can look at the trends among the points in a single grouping. The fact that the data is overlapping is only of minimal significance because it only accepts how far the points are from the means of the Gaussians.

It is also worth noting that both EM and K-Means seemed to do better on the same data sets and worse on the same data sets indicating that the difficultly of optimizing the parameters might be intrinsic to the data in such as a way that both techniques are either helped or hurt.

Acknowledgments

I would like to express my thanks to the my department of Computer Science and Engineering and management of Amity University, Noida for their continuous support and encouragement during this work and also to IJATER Journal for the support to develop this and for their valuable review.

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 SSDBM,pages 21–23, 2004.

[2] K. Bennett and M. Embrechts. An optimization perspective on kernel partial least squares regression.

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

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

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

[12] 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.

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

Biographies

NAMITA BHAN received the B.E. degree in Computer Science and Engineering from the University of Nagpur, Nagpur, Maharashtra, in 2005, pursuing for the M. Tech degree in Computer Science and engineering from the Amity University, Noida, and U.P. Currently, She is a student at Amity University, Noida. Namita Bhan may be reached at
namita.bhan@gmail.com

Prof. (Dr.) Deepti Mehrotra is the Director at Amity School of Computer Sciences, Noida, U.P. and can be reached at
dmehrotra@amity.edu