ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st&22ndMarch, Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

A Resemblance between Credentials in Nptel Application Using Weka Tool

Ms.N.Kalpana , Dr.S.Appavu Alias Balamurugan

Assistant Professor, Dept of Computer Science & Engineering, PSNA College of Engineering & Technology , Dindigul, Tamilnadu, India.

Assistant Professor, Dept of Computer Science & Engineering, PSNA College of Engineering & Technology , Dindigul, Tamilnadu, India.

Professor, Dept of Information & Technology, KLNCIT, Madurai, Tamilnadu, India.

Abstract: Many data mining techniques have been proposed for mining useful patterns in text documents. However, how to effectively use and update discovered patterns is still an open research issue, especially in the domain of text mining. Since most existing text mining methods adopted term-based approaches, they all suffer from the problems of polysemy and synonymy. Over the years, people have often held the hypothesis that pattern (or phrase)-based approaches should perform better than the term-based ones, but many experiments do not support this hypothesis. This paper presents an innovative and effective pattern discovery technique which includes the processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information. Word similarity and Information extraction systems are traditionally implemented as a pipeline of special-purpose processing modules targeting the extraction of a particular kind of information. A fundamental data-mining problem is to examine data for "similar" items. These pages could be plagiarized, for example, or they could be mirrors that have almost the same pleased, but differ in information about the host and about other mirrors. We introduce a technique called "min hashing," which compresses large sets in such a way that we can still deduce the similarity of the underlying sets from their compressed versions. Finally, we explore notions of "similarity" that are not expressible as intersection of sets. This study leads us to consider the theory of distance measures in arbitrary spaces.

Keywords: Clustering, jaccard , min hashing, sensitive hashing , Mixed-Type Attributes

Clustering and classification are fundamental tasks in Data Mining. Clustering and classification are fundamental tasks in Data Mining. Classification is used mostly as a supervised/unsupervised learning method.
The goal of clustering is descriptive, that of classification is predictive [7]. Since the goal of clustering is to discover a new set of categories, the new groups are of interest in themselves, and their assessment is intrinsic. In classification tasks, however, an important part of the assessment is extrinsic, since the groups must reflect some reference set of classes. "Understanding Clustering groups data instances into subsets in such a manner that similar instances are grouped together, while different instances belong to different groups. The instances are thereby organized into an efficient representation that characterizes the population being sampled. Formally, the clustering structure is represented as a set of subsets Consequently, any instance in S belongs to exactly one and only one subset. Clustering of objects is as ancient as the human need for describing the salient characteristics of men and objects, and identifying them with a type [4].

1.1 Similarity of Documents

An important class of problems that Jaccard similarity addresses well is that of finding textually similar documents in a large corpus, such as the Web or a collection of news articles. One should understand that the aspect of similarity is character-level similarity, not "similar meaning," which requires assessment of the words in the documents and their uses. However, textual similarity also has important uses. Many of these involve finding duplicates or near duplicates. First, let us observe

that testing whether two documents are exact duplicates is easy; just compare the two documents character-by-character, and if they ever differ then they are not the same. However, in many applications, the documents are not identical, yet they share large portions of their text.

1.2 Plagiarism

Establishing plagiarism tests our ability to find textual similarity. The plagiarizer may extract only some parts of a document. He may alter a few words and the order of sentences of the original appear. Yet the resulting document may still contain 50% or more of the original. No simple process of comparing documents character by character will detect a sophisticated plagiarism. Given two p-dimensional instances,
xi = (xi1; xi2; : : : ; xip) and
xj =(xj1; xj2; : : : ; xjp),

The commonly used Euclidean distance between two objects is achieved when g = 2. Given g = 1, the sum of absolute paraxial distances (Manhattanmetric) is obtained, and with g=1 one gets the greatest of the paraxial distances (Chebychev metric). The measurement unit used can affect the clustering analysis. To avoid the dependence on the choice of measurement units, the data should be standardized.

1.3 Mirror Pages

It is common for important or popular Web sites to be duplicated at a number of hosts, in order to share the load. The pages of these mirror sites will be quite similar, but are rarely identical. For instance, they might each contain information associated with their particular host, and they might each have links to the other mirror sites but not to themselves. A related phenomenon is the appropriation of pages from one class to another. These pages might include class notes, assignments, and lecture slides. Similar pages might change the name of the course, year, and make small changes from year to year. It is important to be able to detect similar pages of these kinds, because search engines produce better results if they avoid showing two pages that are nearly identical within the first page of results.

1.4 Shingling of Documents

The most effective way to represent documents as sets, for the purpose of identifying lexically similar documents is to construct the set of short strings that appear within it. Then documents that share pieces as short as sentences or even phrases will have many common elements in their sets, even if those sentences appear in different orders in the two documents. In this section, we introduce the simplest and most common approach, called shingling, as well as an interesting variation.

2. Locality-Sensitive Hashing for Documents

Even though we can use min hashing to compress large documents into small signatures and preserve the expected similarity of any pair of documents, it still may be impossible to find the pairs with greatest similarity efficiently. The reason is that the number of pairs of documents may be too large, even if there are not too many documents.

If our goal is to compute the similarity of every pair, there is nothing we can do to reduce the work, although parallelism can reduce the elapsed time. However, often we want only the most similar pairs or all pairs that are above some lower bound in similarity. If so, then we need to focus our attention only on pairs that are likely to be similar, without investigating every pair. There is a general theory of how to provide such focus, called locality-sensitive hashing (LSH) or near-neighbor search. In this section regard as a specific form of LSH, designed for the particular problem we have been studying: documents, represented by shingle-sets, then min hashed to short signatures. We present the general theory of locality-sensitive hashing and a number of applications and related techniques.

2.1 Distance Measures

We now take a short detour to study the general notion of distance measures. The Jaccard similarity is a measure of how close sets are, although it is not really a distance measure. That is, the closer sets are, the higher the Jaccard similarity. Rather, 1 minus the Jaccard similarity is a distance measure; it is called the Jaccard distance. However, Jaccard distance is not the only measure of closeness that makes sense. We shall examine in this section some other distance measures that have applications. Then, in Section 3.6 we see how some of these distance measures also have an LSH technique, that allows us to focus on nearby points without comparing all points. Other applications of distance measures will appear.

2.2 Definition of a Distance Measure

Suppose we have a set of points, called a space. A distance measure on this space is a function d(x, y) that takes two points in the space as arguments and
produces a real number, and satisfies the following axioms:
1. $d(x, y) \geq 0$ (no negative distances).
2. $d(x, y) = 0$ if and only if $x = y$ (distances are positive, except for the distance from a point to itself).
3. $d(x, y) = d(y, x)$ (distance is symmetric).
4. $d(x, y) \leq d(x, z) + d(z, y)$ (the triangle inequality).
The triangle inequality is the most complex condition. Intuitively it implies, that to travel from x to y, we cannot obtain any benefit if we are forced to travel via a third

point z. The triangle-inequality axiom is what makes all distance measures behave as if distance describes the length of a shortest path from one point to another.

2.3 Euclidean Distances

The most familiar distance measure is the one we normally think of as "distance." An n-dimensional Euclidean space is one where points are vectors of n real numbers. The conventional distance measure in this space, which we shall refer to as the L2-norm, is defined:
d([x1, x2, . . . , xn], [y1, y2, . . . , yn]) =vuut
n
Xi=1
(xi − yi)2
That is, we square the distance in each dimension, sum the squares, and take the positive square root.

3. Distance Measures

It is easy to verify the first three requirements for a distance measure are satisfied. The Euclidean distance between two points cannot be negative, because the positive square root is intended. Since all squares of real numbers are nonnegative, any i such that xi*6= yi [check? I have put *] forces the distance to be strictly positive. On the other hand, if xi = yi for all i, then the distance is clearly 0. Symmetry follows because (xi − yi)2 = (yi − xi)2. [exponents?]The triangle inequality requires a good deal of algebra to verify. However, it is well understood to be a property of Euclidean space: the sum of the lengths of any two sides of a triangle is no less than the length of the third side. There are other distance measures that have been used for Euclidean spaces. For any constant r, we can define the Lr-norm to be the distance measure d defined by:
d([x1, x2, . . . , xn], [y1, y2, . . . , yn]) = (Xi=1)

The case r = 2 is the usual L2-norm just mentioned. Another common distance measure is the L1-norm, or Manhattan distance. There, the distance between two points is the sum of the magnitudes of the differences in each dimension.

It is called "Manhattan distance" because it is the distance one would have to travel between points if one were constrained to travel along grid lines, as on the streets of a city such as Manhattan. Another interesting distance measure is the L∞-norm, which is the limit as r approaches infinity of the norm. As r gets larger, only the dimension with the largest difference matters, so formally, the L∞-norm is defined as the maximum of |xi − yi| over all dimensions

3.1 Distance Measures for Binary Attributes

The distance measure described in the last section may be easily computed for continuous-valued attributes. In the case of instances described by categorical, binary, ordinal or mixed type attributes, the distance measure should be revised. In the case of binary attributes, the distance between objects may be calculated based on a contingency table. A binary attribute is symmetric if both of its states are equally noteworthy In that case, using the simple matching coefficient can assess dissimilarity between two objects:
d(xi; xj) = r + s
q + r + s + t
where q is the number of attributes that equal 1 for both objects; t is the number of attributes that equal 0 for both objects; and s and r are the number of attributes that are unequal for both objects. A binary attribute is asymmetric, if its states are not equally important (usually the positive outcome is considered more important). In this case, the denominator ignores the unimportant negative matches (t). This is called the Jaccard coefficient:

d(xi; xj) = r + s

q + r + s

3.2 Distance Measures for Nominal Attributes

When the attributes are nominal, two main approaches may be used:
1. Simple matching:
d(xi; xj) = p ¡ m p

where p is the total number of attributes and mis the number of matches.

2. Creating a binary attribute for each state of each nominal attribute and computing their dissimilarity as described above.

4. Similarity Functions

An alternative concept of the distance is the similarity function s(xi; xj) that compares the two vectors xi and xj. This function should be symmetrical (namely s(xi; xj) = s(xj ; xi)) and have a large value when xi and xj are somehow "similar" and constitute the largest value for identical vectors. A similarity function where the target range is [0,1] is called a dichotomous similarity function. In fact, the methods described in the previous sections for calculating the "distances" in the case of binary and nominal attributes may be considered as similarity functions, rather than distances.

4.1 Evaluation Criteria Measures

Evaluating if a certain clustering is good or not is a problematic and controversial issue. In fact Bonner was the first to argue that there is no universal definition for what is a good clustering. The evaluation remains mostly in the eye of the beholder. Nevertheless, several evaluation criteria have been developed in the literature. These criteria are usually divided into two categories:Internal and External

II.RELATED WORK

The author [6] described a framework for generating an approximate top-k answer, with some probabilistic guarantees. In our work, we use the same idea. The main and crucial difference is that we only have "random access" to the underlying database (i.e., through querying), and no "sorted access.". The author [6] assumed that at least one source provides "sorted access" to the underlying content . He describes sampling strategies for estimating the relevance of the documents retrieved by different keyword queries [3][4].

III.IMPLEMENTATION FRAMEWORK

Preprocess:

- Load Data
- Preprocess Data
- Analyze Attributes

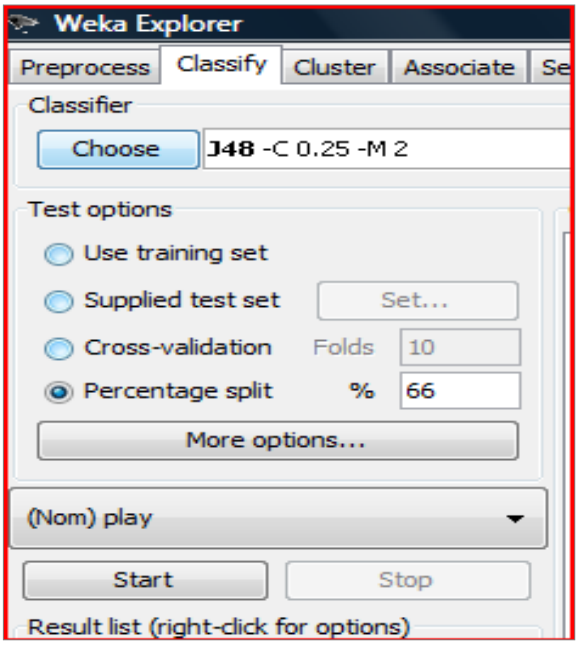
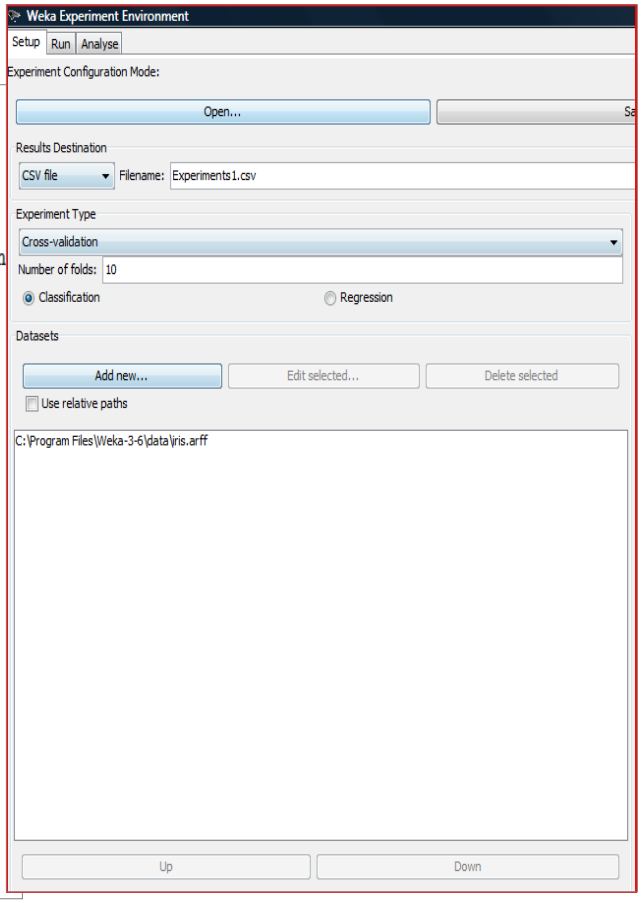


Fig 1

More Classifiers:

- `trees.J48` A clone of the C4.5 decision tree learner
- `bayes.NaiveBayes` A Naive Bayesian learner. `-K` switches on kernel density estimation for numerical attributes which often improves performance.
- `meta.ClassificationViaRegression -W functions.LinearRegression` Multi-response linear regression.
- `functions.Logistic` Logistic Regression.
- `functions.SMO` Support Vector Machine (linear, polynomial and RBF kernel) with Sequential Minimal Optimization Algorithm due to [3]. Defaults to SVM with linear kernel, `-E 5 -C 10` gives an SVM with polynomial kernel of degree *5* and lambda of *10*.
- `lazy.KStar` Instance-Based learner. `-E` sets the blend entropy automatically, which is usually preferable.
- `lazy.IBk` Instance-Based learner with fixed neighborhood. `-K` sets the number of neighbors to use. `IB1` is equivalent to `IBk -K 1`
- `rules.JRip` A clone of the RIPPER rule learner.



.Fig 2

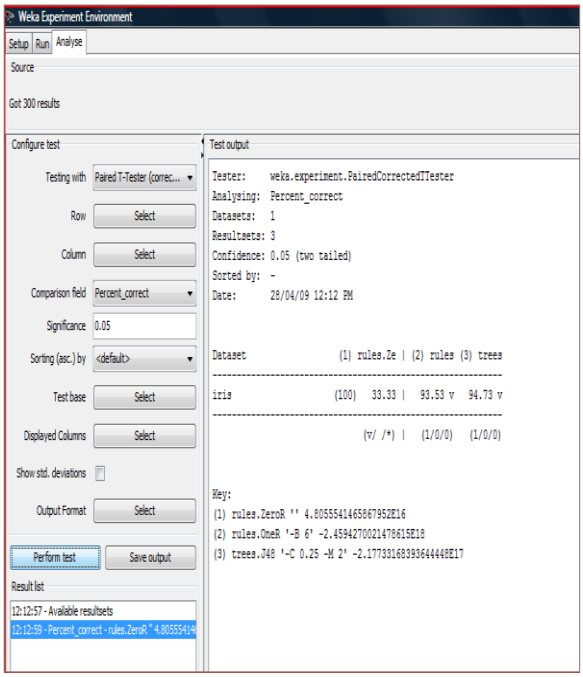


Fig 3

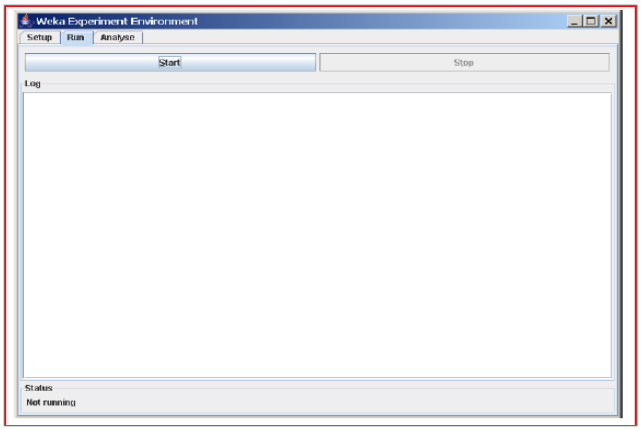


Fig 4

IV.CONCLUSION

Finally, we investigated the crisis of content based similarity in NPTEL application, which is essentially a distributed storage system. To ensure the Extraction of users' data in repository, we proposed an effective and flexible search scheme with explicit dynamic data support, including character, word, and phrase base similarity measure. We rely on content in the file preparation to provide redundancy parity vectors and guarantee the data dependability.

REFERENCES

[1] I.F. Ilyas, G. Beskales, and M.A. Soliman, "A Survey of Top-K Query Processing Techniques in Relational Database Systems," ACM Computing Survey, vol. 40, no. 4, pp. 1-58, 2008.

[2] A. Penev and R. K. Wong. Finding similar pages in a social tagging repository. In WWW, pages 1091–1092, 2008.

[3] A. Telang, C. Li, and S. Chakravarthy. One size does not fit all: Towards user- and query-dependent ranking for web databases. Technical report, UT Arlington, http://cse.uta.edu/research/Publications/Downloads/CSE- 2009-6.pdf, 2009.

[4] V. Hristidis, Y. Hu, and P.G. Ipeirotis, "Ranked Queries Over Sources with Boolean Query Interfaces without Ranking Support," Proc. 26th IEEE Int'l Conf. Data Eng. (ICDE '10), 2010.

[5] A. Telang, S. Chakravarthy, and C. Li. Establishing a workload for ranking in web databases. Technical report, UT Arlington, http://cse.uta.edu/research/Publications/Downloads/CSE-2010- 3.pdf, 2010.