IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12, Issue 1, 2024

Incremental Data Stream Mining and Conflict Analysis for the Recognition of Sonar Signals Underwater

Koteswararao Kodali [1],Raamagi Vinod Kumar [2,] Kasturi Bharathi [3],

Associate Professor[1], Assistant Professor [2,]UG Students[3,]

Department of CSE

BRILLIANT GRAMMAR SCHOOL EDUCATIONAL SOCIETY'S GROUP OF INSTITUTIONS-INTEGRATED CAMPUS Abdullapurmet (V), Hayath Nagar (M), R.R.Dt. Hyderabad.

Introduction

Sound navigation and range, or sonar, is a sound propagation technique that is used for underwater navigation, communication, and/or detection of sub marine objects. According to [1], the relevant methodologies have recently been examined. The detection and categorization of sonar sounds, in particular, was cited as one of the most difficult problems in the area. For the detection of items of interest beneath the sea, selecting the proper categorization model for sonar signals identification is critical. Ocean sampling networks, environmental monitoring, offshore exploration, catastrophe prevention, aided navigation, and mine reconnaissance may all benefit from underwater sensor networks. Sensor networks in the sea are simple to set up, don't need any connections, and don't get in the way of shipping. However, long-distance underwater sonar transmissions are susceptible to interference and noise. As a result, sonar signal identification uses data mining methods such as classification to identify the target object's surface from which sonar waves are reflected [3–5]. It is possible to gain significant accuracy in conventional data mining by utilising the whole dataset to build a classification model. Although the induction is normally done and repeated in batches, this means that the model's accuracy is likely to degrade between updates [6]. When new data is added to a dataset, the update time may increase as the total amount of data grows. Sonar signals are relentless and continuously detected, much like any other data stream. Batch mode classification techniques, despite their accuracy, may not be ideal for streaming applications like sonar sensing. It is critical for real-time sensing and reconnaissance to make data processing times as fast as possible since sonar signal data streams might potentially add up to infinity. Using quick conflict analysis from the stream-based training dataset, we provide an alternative data stream mining approach for progressively purging noisy data. With the use of conflict analysis (iDSM-CA), it is known as an incremental data stream mining technique (in acronym). An benefit of this strategy is that it can progressively develop a classification model from stream data. Simulation tests are used to demonstrate the effectiveness of the suggested technique, particularly when it comes to reducing noise from the sonar data while it is streaming. What follows is an outline of the remainder of the paper. Section 2 provides an overview of some of the most often used computational methods for removing noise from training datasets. The "conflict analysis" technique for deleting incorrectly categorised occurrences is described in Section 3 of our novel data stream mining approach. A set of sonar recognition tests are presented in Section 4 to verify the stream mining technique. The paper comes to a close in Section 5.

Related Work

Researchers have tried a variety of methods to identify and remove noise in the training dataset, which is commonly referred to as random chaos. In essence, these methods look for data examples that throw off the training model and reduce the accuracy of the classification. The main focus is on identifying data anomalies and figuring out how they impact categorization accuracy. There are three main groups of these methods: statistical, similarity-based and categorization. This section focuses on statistical methods for noise detection. Data with very high values are treated as noise in this kind of analysis. From simple normality tests to discovering extreme values beyond a particular number of standard deviations, there are a wide variety of detection strategies to choose from in the scientific literature. [7, 8] provide comprehensive reviews of outlier detection algorithms used to find noise in preprocessing. According to [9], the authors used a

novel outlier identification technique that relies on a dataset's predicted behaviour. A sparse point in the lower low-dimensional projection is regarded abnormal and is eliminated from the data. Determined via brute force or at best a heuristic method are the projections. Clustering characteristics may be found on both nonleaf nodes and leaf nodes in a similar approach discussed in [10]. As a result, the leaf nodes with low density are filtered out. Methods for detecting noise based on similarity Most of these approaches need the use of a standard against which the experimental results can be compared in order to determine how similar or different they are. Before looking for the subset that would have the largest impact on the training dataset's dissimilarity, the researchers in [11] separated their data into many subgroups. Variance is an example of a dissimilarity function, which returns a low value for similar items and a high value for dissimilar elements in the same dataset. It's difficult to establish a universal dissimilarity function, though, the authors said. Hyperclique-based data cleaner HCleaner was suggested by Xiong et al [12]. There is a high degree of resemblance between every pair of items in a hyperclique pattern because of the intensity of the association between the two occurrences. When an instance is omitted from any hyperclique pattern, the HCleaner considers it "noise." Using a k-NN method, another group of researchers [13] found that outliers may be identified by comparing test data to nearby data. Different data are handled as erroneously categorised instances and deleted as a result of utilising their closest neighbours as references. As a result of their research into data patterns, the authors came up with Wilson's editing approach: a set of rules for selecting and erasing certain data sets on the fly. Noise Detection Based on Classification. Preliminary classifiers are used to help determine whether data instances are improperly categorised and should be deleted using classification-based procedures. An n-fold cross validation technique was employed in [14] to detect mislabeled cases. The dataset is divided into n subgroups using this method. Omitted instances are assigned to one of two classes: one for those that have been excluded from training, and one for those that have been included. If an instance is misclassified by any of the classifiers, it is flagged as such. Filtering may be accomplished with a simple majority vote or a more involved consensus procedure. This is not the first time academics have come up with a way to remove outliers. The training dataset is used to build a pruning tree, which is then utilised to categorise the data. Those instances from the training dataset that the pruned tree erroneously classifies are eliminated. To ensure that the pruned tree can properly classify every occurrence in the training dataset, these steps are repeated. Genetic algorithm (GA) was used to construct a collection of suspect noisy examples and pick one of the prototypes to identify the real set of noisy instances in the work published in [16]. The GA's fitness function is a pre-built generic classifier that it utilises to look for instances of incorrect classification.

Our Proposed Data Stream Mining Model

If you're looking for a way to remove unwanted data from a large amount of data, you'll need to use the strategies above. In contrast to the methods described in Section 2, the one provided here for data preparation and model learning is entirely new. This preprocessing procedure has traditionally been seen as a separate phase that precedes model learning. Once the information has been scoured, it is determined which occurrences need to be omitted since they might lead to incorrect classifications in the near future. It is possible to filter the training set

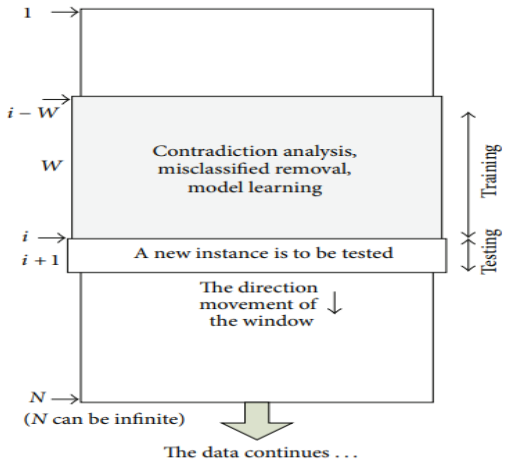


figure 1: Illustration of how iDSM-CA works.

subsequently included into the learning process in the hope that it would reduce the noise. To the contrary, the incremental learning process of iDSM-CA incorporates the detection of noise as well as the removal of data that has been incorrectly categorised. As the data stream comes in, preparation and training work is followed by testing work. Figure 1 depicts the motion of a window of size W along a data stream. A conflict analysis (noise detection) is carried out initially, followed by the elimination of incorrect

ISSN 2347–3657

Volume 12, Issue 1, 2024

data and the training of new employees (model building). Incoming examples are tested when the model has been properly trained. To compute the average level of performance based on overall results, intermediate results may be generated using this method, and the overall performance results can also be used. Processing and incremental learning model workflow in 3.1. Figure 2 depicts the iDSM-complete CA's operating procedure. Data processing and training are carried out simultaneously inside a single window, which slides along the data stream from the beginning. Anytime technique is a data mining term that refers to models that may be used immediately without waiting for all of the training data to be collected. When fresh data is received, the window gradually covers the older (outdated) instances and fades them out. Real-time updates are made to the model when the analysis begins again. As a result of this method, the benefits of deleting misclassified instances are realised without having to presume that the dataset is static and limited. The training dataset enclosed in the window W is continuously improved and the model gradually learns from each new piece of data as the window moves forward.

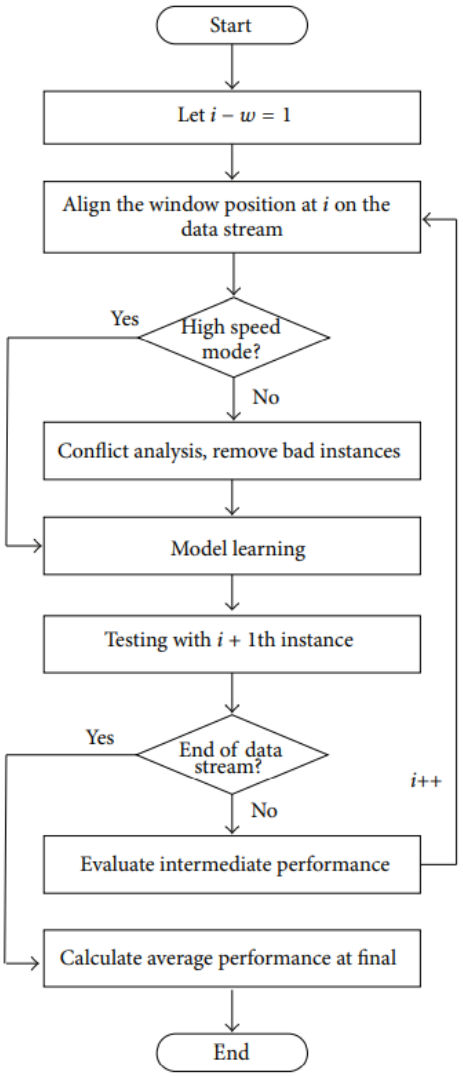


Figure 2: Workflow of the incremental learning method.

W.'s incorporation of new data The rolling window W's statistics may be cumulative, which is another advantage of the suggested technique. The properties of the data are discreetly recorded from a long-term global viewpoint by collecting statistics on the contradiction analysis carried out inside each frame of the window as it advances. With the use of global information, it may be possible to increase the accuracy of the analysis of noisy data. To put it another way, the noise detection algorithm becomes better at picking up noise as it gains experience (by using cumulative data). It's important to remember that noise is a relative term that can only be properly

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12, Issue 1, 2024

understood in the context of a pre-existing standard. 3.2. The Analysis of Conflict. A modified pair-wise-based classifier (PWC) based on the interdependence of attribute values and class labels is employed for contradiction analysis. Similar to an instance-based classifier, PWC only gets triggered when testing an instance and progressively learns at most one round a classifier.

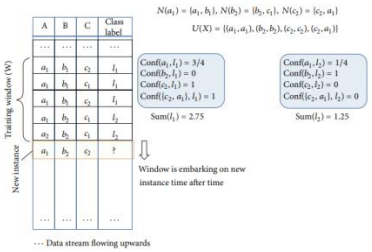


Figure 3: Illustration of a dynamic rolling window and conflict analysis.

PWC provides a number of advantages over other preprocessing approaches in addition to its speed, which is essential for light preprocessing. While computing the supports and confidence values is all that is required to estimate which target label an instance belongs to, there is no need to retain a persistent tree structure or trained model beyond small registers for statistical purposes. Additionally, the samples (references) required for noise detection can scale up or down to any amount (W). Figure 3 is an example of a weighted PWC based on [17]. The sliding window comprises W possible training samples spanning three characteristics (A, B, and C) and one target class for each ith iteration of incremental model updating. As an example, X contains a vector of values a1, b2, and c2 at position i+1, which is right before the previous end of the window. Because we're going to assume that k equals W/2, we can see the neighbour sets for each attribute's X values in the figure's upper-right corner. Because conf (a1, b1) = 0.75 and conf (a1, a1) = 1 are the greatest two values for a1, N(a1) = a1, b1 A list of the resultant U(X) sets may be found here. For example, in U(X) only a1 is connected with a1, producing the pair (a1, a1). Despite the fact that b1 N(a1), it has no place in X and should be omitted from U(x). This also applies to c1 in relation to N(b2), while U(X) connected with c1 includes c2 and a1, which are also part of N(c2). PWC analyses the confidence levels for each member of U(X) against the two target classes l1 and l2. To test first-order dependence between (a1 and l1), we use the following formula: (a1/l1) = support(a1, l1)/support(a1) = 3/4 = 0.75. However, given the pair (c2, a1), we evaluate second-order dependence by doing the following calculation: Sum (l1) = 2.75 and Sum (l2) = 1.25, which means that the new instance belongs to class l1 based on this calculation. By comparing the computed class membership to the class label for the new instance, conflict is identified. There is no conflict assumed if a new instance is in agreement with the PWC computation, and the window goes ahead by incrementing one row, excluding the last row, and incorporating the new instance in the training set. The new instance is purged if the class label of the new instance conflicts with the result of the computed class label. Neighbor sets are one of the changes we made to our methodology. Only the most recent confidence values of the pairings inside the current window frame are stored in the current neighbour sets, which are referred to as Local Sets. Local Sets are updated every time a new instance is added to a window, thus the old information is wiped out and replaced. A comparable buffer called Global Sets is employed in our approach that does not replace but accumulates the freshly calculated confidence values for each pair in the window. Yes, conflict analysis can be done using comparable methods. By using PWC, it can be observed that the amount of information and computation necessary is reduced to a minimum, resulting in faster operation.

Experiment

The experiment's goal is to test the suggested iDSM-CA approach for underwater sonar signal detection. iDSM-CA will be tested in a data stream mining scenario in contrast to existing data mining approaches for detecting sonar sound data. iDSM-CA was used to assess the sonar identification capabilities of six different classification algorithms. Two typical algorithms will be used for classic batch-based learning.

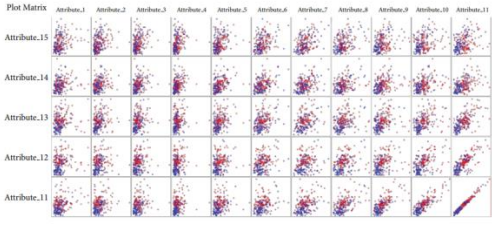


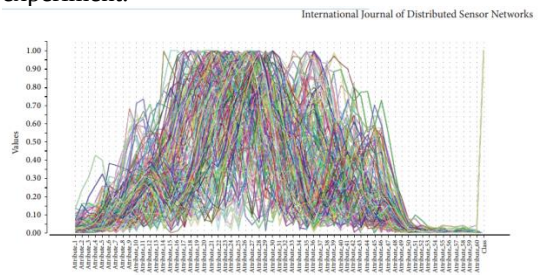
Figure 4: Matrix visualization of data over multiple attributes of the sonar dataset.

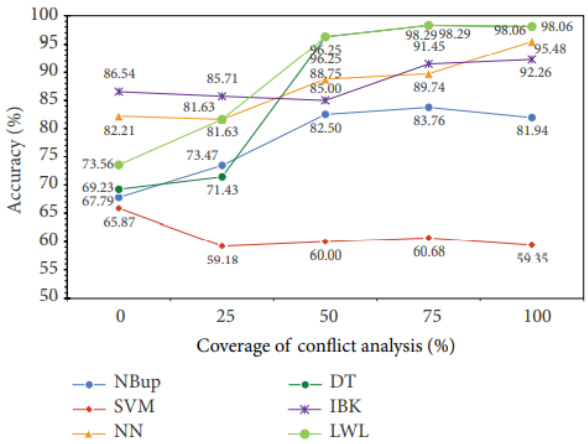
International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12, Issue 1, 2024

Neural networks (NNs) and support vector machines (SVMs) (SVM). Decision tables (DT), K-nearest neighbours classifiers (IBK), and locally weighted learning are examples of instance-based classifiers for iDSM-CA (LWL). An incremental variant of the technique, dubbed updateable naive Bayesian (NBup), is presented as well for intellectual interest. In general, batch learning and incremental learning are both possible with all six methods. Models are trained and updated section by section in incremental learning mode, with conflict analysis being enforced at all times. The trained model is tested using the next fresh data instance that comes along as the window moves. From the beginning to the finish, the performance data are gathered. It's common practise to train a model using a whole batch of data before assessing its performance using a 10-fold validation method. Java-based software is used for the experiment.





Under iDSM-CA, sonar classification accuracy is shown in Figure 6.

It typically suggests that the qualities and classes have complicated and highly nonlinear relationships. It would be difficult for a classifier to achieve high recognition accuracy if this was the case. Classification model prediction accuracy, model training duration, and ROC indices were all evaluated using this data in our experiment. Time spent to develop a full classifier utilising all of the data in batch learning is measured as model training. There are three stages of data processing, conflict analysis, and incremental training that all contribute to a model's learning time: the data processing stage, conflict analysis stage, and incremental training stage (the mean time per sliding step). True classification is defined as the percentage of cases that have been accurately categorised. As an indicator of the test's capacity (which is the strength of discriminating in classification) to distinguish between alternative states of target objects, the ROC serves as a unified degree of accuracy that ranges from 0 to 1. Random guessing is just as accurate as using the model's ROC. Starting with a calibration step in which six algorithms are used to determine the appropriate size of W, we begin the experiment. With just a modest sample size and occasional calibrations or when the incremental learning performance diminishes, fine-tuning the window size may be done. Windows are gradually increased in size from 49 to 155 in our experiment. The different window widths are labelled as 25%, 50%, 75%, and 100% with proportion to the whole dataset as a simple convention. Full batch learning is comparable to incremental learning with W=0%. Model induction time (Figure 7), classification accuracy (Figure 6), and ROC index (Figure 8) are all shown in the following graphs for the six classification algorithms that were tested using the iDSM-CA. A straightforward measure of the classifier's ability to distinguish between rock and metal is accuracy, expressed as a percentage. This means that the algorithms used in a data stream mining environment must be able to generate models in a timely manner. When updating/refreshing the model on the fly, algorithms should take as little time as possible. The dataset will not be cleaned up if the window size W=0 indicates there is no conflict analysis being used. Classification models' accuracy varies depending on the method used and the size of the sliding window. Figure 6 shows that the incremental group of algorithms outperforms the conventional group of algorithms in terms of classification accuracy.

Conclusion

Sonar signal detection is recognised to be a difficult subject, yet it has a substantial military impact. Noise in the undersea environment is a key contributor to inaccurate readings. Noise interferes with the creation of classification models, causing confusion. Classification models are disrupted and metaknowledge is distorted by erroneous rules in classification models due to inconsistent data in training datasets that do not accord with the bulk of the data; this leads to erroneous rules in classification models and distorts training patterns. Outliers, misclassified cases, and misfits are all terms used by other authors to describe noise, all of which are data categories whose removal would enhance the classification model's accuracy. It has been researched for over two decades, however methods previously described to remove this kind of noise presuppose batch processes and the use of the whole dataset for noise identification..iDSM-CA, which stands for incremental data stream mining with conflict analysis, was described in this study. The iDSM-lightweight CA's window-sliding mechanism, which is intended to mine moving data streams, is its primary benefit. The iDSM CA model is a lot easier to use than some of the more complicated strategies mentioned in Section 2 of this document. Its great speed and effectiveness in supplying a noise-resilient training dataset for incremental learning, employing empirical sonar data in discriminating metal or rock items, have been shown in our experiment. Experiments have shown the efficacy and efficiency of iDSM-CA in mining stream data. The ability to analyse live data streams is critical in a variety of different types of big data applications, such as data mining sports activities [19] and social media data feeds [20].

Acknowledgments

Grant no. MYRG073(Y3-L2)-FST12-FCC, awarded by the University of Macau, FST, and RDAO, is gratefully acknowledged by the authors for their financial assistance. Adaptive OVFDT with Incremental Pruning and ROC Corrective Learning for Data Stream Mining

References

[1] H. Peyvandi, M. Farrokhrooz, Roufarshbaf, and S.-J. Park, “SONAR systems and underwater signal processing: classic and modern approaches,” in Sonar Systems, N. Z. Kolev, Ed., pp. 173– 206, InTech, Hampshire, UK, 2011.

[2] C. Lv, S. Wang, M. Tan, and L. Chen, “UA-MAC: an underwater acoustic channel access method for dense mobile underwater sensor networks,” International Journal of Distributed Sensor Networks, vol. 2014, Article ID 374028, 10 pages, 2014.

[3] H. Akbarally and L. Kleeman, “Sonar sensor for accurate 3D target localisation and classification,” in Proceedings of the 1995 IEEE International Conference on Robotics and Automation, pp. 3003–3008, May 1995.

[4] A. Heale and L. Kleeman, “Fast target classification using sonar,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 1446–1451, November 2001.

[5] A. Balleri, Biologically inspired radar and sonar target classification [Ph.D. thesis], University College London, London, UK, 2010.

[6] S. Fong, H. Yang, S. Mohammed, and J. Fiaidhi, “Stream-based biomedical classification algorithms for analyzing biosignals,” Journal of Information Processing Systems, Korea Information Processing Society, vol. 7, no. 4, pp. 717–732, 2011.

[7] X. Zhu and X. Wu, “Class noise vs. attribute noise: a quantitative study,” Artificial Intelligence Review, vol. 22, no. 3, pp. 177–210, 2004.

[8] V. J. Hodge and J. Austin, “A survey of outlier detection methodologies,” Artificial Intelligence Review, vol. 22, no. 2, pp. 85–126, 2004.

[9] C. Aggarwal and P. Yu, “Outlier detection for high dimensional data,” in Proceedings of the ACM SIGMOD International Conference [9] C. Aggarwal and P. Yu, “Outlier detection for high dimensional data,” in Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 37–46, May 2001. 12 International Journal of Distributed Sensor Networks

[10] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: an efficient data clustering method for very large databases,” in Proceedings of the Conference of Management of Data (ACM SIGMOD ’96), pp. 103–114, 1996.

[11] A. Arning, R. Agrawal, and P. Raghavan, “A linear method for deviation detection in large databases,” in Proceedings of the 1996 International Conference on Knowledge Discovery and Data Mining (KDD ’96), pp. 164–169, Portland, Ore, USA, 1996.

ISSN 2347–3657

Volume 12, Issue 1, 2024

[12] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, “Enhancing data analysis with noise removal,” IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 3, pp. 304–319, 2006.

[13] H. Brighton and C. Mellish, “Advances in instance selection for instance-based learning algorithms,” Data Mining and Knowledge Discovery, vol. 6, no. 2, pp. 153–172, 2002.

[14] C. E. Brodley and M. A. Friedl, “Identifying and eliminating mislabeled training instances,” in Proceedings of the 1996 13th National Conference on Artificial Intelligence (AAAI ’96), pp. 799–805, AAI Press, Portland, Ore, USA, August 1996.

[15] G. H. John, “Robust decision tree: removing outliers from databases,” in Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, pp. 174–179, 1995.

[16] B. Byeon, K. Rasheed, and P. Doshi, “Enhancing the quality of noisy training data using a genetic algorithm and prototype selection,” in Proceedings of the 2008 International Conference on Artificial Intelligence (ICAI ’08), pp. 821–827, Las Vegas, Nev, USA, July 2008.

[17] A. Nanopoulos, A. N. Papadopoulos, Y. Manolopoulos, and T. Welzer-Druzovec, “Robust classification based on correlations between attributes,” International Journal of Data Warehousing and Mining, vol. 3, no. 3, pp. 14–27, 2007.

[18] R. P. Gorman and T. J. Sejnowski, “Analysis of hidden units in a layered network trained to classify sonar targets,” Neural Networks, vol. 1, no. 1, pp. 75–89, 1988.

[19] I. Fister Jr., I. Fister, D. Fister, and S. Fong :, “Data mining in sporting activities created by sports trackers,” in Proceedings of the 2013 International Symposium on Computational and Business Intelligence (ISCBI ’13), pp. 88–91, 2013.

[20] J. Fiaidhi, O. Mohammed, S. Mohammed, S. Fong, and T. H. Kim, “Mining twitterspace for information: classifying sentiments programmatically using Java,” in Proceedings of the IEEE 7th International Conference on Digital Information Management (ICDIM ’12), pp. 303–308, Taipa, Macau, August 2012