https://doi.org/10.5392/IJoC.2019.15.3.032

Design and Implementation of Incremental Learning Technology for Big Data Mining

Byung-Won Min and Yong-Sun Oh
Division of Information and Communication Convergence Engineering
Mokwon University, Daejeon, 35349, Korea

ABSTRACT

We usually suffer from difficulties in treating or managing Big Data generated from various digital media and/or sensors using traditional mining techniques. Additionally, there are many problems relative to the lack of memory and the burden of the learning curve, etc. in an increasing capacity of large volumes of text when new data are continuously accumulated because we ineffectively analyze total data including data previously analyzed and collected. In this paper, we propose a general-purpose classifier and its structure to solve these problems. We depart from the current feature-reduction methods and introduce a new scheme that only adopts changed elements when new features are partially accumulated in this free-style learning environment. The incremental learning module built from a gradually progressive formation learns only changed parts of data without any re-processing of current accumulations while traditional methods re-learn total data for every adding or changing of data. Additionally, users can freely merge new data with previous data throughout the resource management procedure whenever re-learning is needed. At the end of this paper, we confirm a good performance of this method in data processing based on the Big Data environment throughout an analysis because of its learning efficiency. Also, comparing this algorithm with those of NB and SVM, we can achieve an accuracy of approximately 95% in all three models. We expect that our method will be a viable substitute for high performance and accuracy relative to large computing systems for Big Data analysis using a PC cluster environment.

Key words*: Incremental Learning, Classifier, Classification Scheme, Big Data Mining, Re-learn, Feature*(*s*).

1. INTRODUCTION

Recently technology of real-time data processing becomes more important as Big Data environment is newly changing along the schemes of WSN(wireless sensor network), smart grid, and other data oriented developments [1]-[3].

We can obtain pretty good solutions of almost real-time cluster parallel processing from 'Impala Apache Tez' of Cloudera or 'Presto' of Facebook in the area of real-time data processing [4], [5]. We had mainly focused on the technologies for storage and management of data using DBMS in the past. Nowadays, however, technology developments are focusing on real-time data processing and their applications getting along with the business environment changes [6], [7]. Following this change of environment, one of the most dominant problems in real-time data processing world is that we don't have any efficient classifying technology for these accumulated streaming data [8]. In addition, embedded DBMS gets to connect other devices to public GIS and therefore social demands of real-time sensor data processing should be dramatically increased [9], [10]. On the other hand, as following this Big Data era, demands of structured and/or non-structured data analysis and time series analysis are truly expanded [11].

It is one of the principal difficulties to process Big Data generated from various digital media and/or sensors using traditional mining methods. Moreover, they have a traumatic inefficiency that they have to re-process the total data when any kind of new data are processed and accumulated, which arises a lot of redundancy and a serious time consumption. Incremental learning is a good advanced technology which learns only additional data or increments to overcome these difficulties of the traditional mining methods in this active environment with frequently added huge amount of data [12].

In the existing studies, they use the feature-selection method for an efficient text processing, which is well known to be necessary for performance enhancement. However computing time and resources required in the procedure of feature characteristics analyses and rejections are indispensable in a Big Data environment in which huge of real-time data are continuously flooding. Traditional feature-selection and reduction methods, therefore, should have a limitation of applications.

The incremental learning technology presented in this paper can only learn incremented part of data and merge the result on the current stuck of data without any repetition of total

* *Corresponding author, Email: ysunoh@mokwon.ac.k*
Manuscript received Jul. 12, 2019; revised Aug. 08, 2019; accepted Aug. 14, 2019

data re-analysis in dynamic creation environment of these raw data. Throughout this improved process of data learning, we can solve the problems for the lack of memory and learn time burden in this learning process of huge amount of data. It doesn't depend on the feature-reduction method in the process of learning data to be free to learn only changed elements when partial variance of characteristics are occurred. We also design and implement a general-purpose classifier applying the concept of incremental learning [13], [14].

This paper consists of the following contents; After this introductory section, we present the concept of incremental learning model and its classifiers in Section 2. In Section 3, we explain the procedure to design and realize an explicit interface for our incremental learning system including the large-party classifier. Section 4 contributes to evaluate the performance of the system. For this purpose, we submit an experimental data set and fulfill the macro averaged test as well as the micro averaged test. Finally we insist our incremental learning technology shows a better performance in structured and/or non-structured data processing and offers a more convenient user environment on distributed parallel processing frameworks in Section 5. And we express our concluding remarks in Section 6.

2. CONCEPT OF INCREMENTAL LEARNING MODEL

2.1 Generation of Elementary Classifiers and Their Dynamic Integration Method

We frequently meet a situation of learning and analyzing over a few millions of information resources when we apply automatic classification technology to this current real service applications. It is well known that we need to use feature-selection method for an efficient processing of text, which results in some reduction of the amount of information as well as performance improvement. Under a Big Data environment, however, feature analysis, selection, reduction and delete will be big burdens for their long processing time and lots of computing resources. And we also have a limitation to choose feature-selection and reduction methods.

To overcome these difficulties, our incremental learning technology generates a lot of small size matrix and dynamically integrates them without any information loss. Fig.1 shows an example of combining several learning results(classifiers) into a large-party model by database level. Applying these process repeatedly, we can create the final version of large scale classifier.

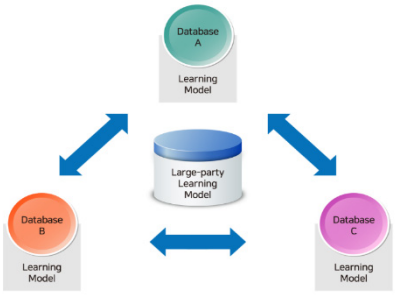


Fig.1 Generation of a Large-Party Classifier

2.2 Generation Process of Elementary Classifiers

We need a series of processes to generate an elementary classifier including the pre-processing. The first step is feature-extraction. We can refer two different types of retrieval to extract features. For the case of information extraction from the 'title' or 'abstract', we create a feature set by stemming (English) or analysis of structure elements(Korean). We had to refer to delete low frequency features in this extraction process. In general, specific low frequency features which occurs only one time in the total text(collection frequency=1) are about 40~60% of all features. On the other hand, we can use the 'key-words' or 'descriptor' field for our extraction. In this process, we can retrieve more important information including 'noun phrases' from non-structural data such as 'title' or 'abstract'.

The second step is to assign category codes on individual features which compose the text. In this step, we create fields such as Unique ID of the text, Feature, and Category code etc.

The third step is to create the characteristic matrix of features and its information for calculation of feature-vectors. We load this matrix as an element of DB or just a binary file. In this paper, we call this information matrix to create an elementary classifier as 'characteristic matrix of feature'. And we create fields such as Unique ID of the feature, Feature itself, Category code, TP, TN, FP, FN, and IDF. Table 1 shows that these features and their range codes are closely related to their appearances and affiliates.

Table 1. Relationship between Feature and Category

Appearance	In Category c_j	Not in Category c_j
Feature f_i shows	*TP*(*True–Positive*)	*FN*(*False–Negative*)
Feature f_i doesn't show	*FN*(*False–Negative*)	*TP*(*True–Positive*)

2.3 Generation of Large Scale Classifier throughout Integrating of Elementary Classifiers

The core of large-party classifier generation is a dynamic integration of elementary classifiers by means of combining 'characteristic matrix of feature' at the 3rd step of the generation framework of elementary classifier. We can create a huge matrix from lots of elementary classifiers through this dynamic integration process following automatic classifications when we got a lot of texts that should be learned.

2.3.1 Dynamic Integration of Matrix

First of all, let some 'characteristic matrix of feature' loaded on memory and make a set of matrix of which unique distinct feature value appears. And bring the information into individual features referring to the objective matrix. At the same time, matrix information such as number of features, number of total texts etc. are dynamically measured and TP(True Positive), TN(True Negative), FP(False Positive), FN(False Negative) etc. should be re-calculated because all the characteristic matrix do not contain all the features. Throughout these processes, the integrated matrix combining 10 classifiers each contains 100,000 learning objects has exactly the same number of individual parametric elements as the matrix of classifier which contains one million learning objects learned simultaneously.

2.3.2 Generation of Feature-Weight Vector for Individual Feature

We load feature vectors on DB or as a binary file. These vectors are created as an appropriate form for voting classifier using distance factor, cosine, LOR(log odds ration) etc. extracted from the characteristic matrix of integrated feature. Feature vector can be expressed as follows;

$$SIM(f,c) = \left[(1 + logTF) * \log\left(\frac{N}{DF}\right)\right] * LOR(f,c) * AM(f,c) \quad (1)$$

In this learning model, we additionally apply OR(odds ratio) and AM(ambiguity measure) models for assigning feature weights.

2.3.3 Performing Text Categorizations

At the next step, we perform a classification by this voting style methodology using feature vectors generated from the integrated matrix mentioned above. FVC(feature voting classifier) is a probabilistic model which has a good classifying performance and speed [15]. After loading the feature vectors created by the method described in the last sub-section on the memory, we can classify huge input texts by this high-speed performance classifying technology.

$$decision[c_j] = argmax_{c_j \in Category} \sum_i SIM(f_i, c_j) \quad (2)$$

The final version of classifier has a relatively small size of data so that it can take a less amount of main memory. Therefore we can obtain a high-speed classifier that fulfill a linear combination of each weighted vectors without suffering from speed degrading by the increment of number of features.

3. DESIGN AND IMPLEMENTATION

3.1 Design

Automatic classification technology based on the incremental learning model proposed in this paper consists of 3 sequential steps such as data collection, text learning & classification, and structure analysis & visualization.

Component of data collection gathers and stores large amount of academic or technological papers. Classifying information are assigned on every paper for automatic performance evaluation of the classification. One part of collected data will be used for learning data set, and the other part for performance evaluation.

Component of text learning & classification performs studying and classifying the huge text collected above throughout automatic classifier. FVC(feature value voting classifier) used in this paper is a kind of probabilistic classifiers, in which the result of classification will be a similarity of probability between the subject and the range. We can, therefore, gather text data most ambiguously classified by the system using this probability measure. Results from this procedure can be easily advanced as a quality improvement system of text data throughout active learning.

The final component is for the visualization throughout intellectual structure analysis. Using the error-log generated by automatic classification, we can compose a total network of learning fields with the similar method of measuring similarity between branches using probabilistic similarity coefficients. Therefore the total system is designed to be operating in a wormlike manner of each component because the system analyzes the network structure using this result of automatic classification.

3.2 Implementation

In this subsection, we present the design process and its implementation of the system we proposed here. The system developed in this paper includes a convenient web-based interface to be easily approached by users and can be operated under any kind of OS environments like Windows and Linux etc. In addition we implement an explicit interface so that any user can operate classification tasks without any expert knowledge. Fig.2 shows our display of the Web-based integrated interface.

Fig.2 Screen-Shots of Web-Based Integrated Interface

We develop work and resource management scheme so that we can easily perform incremental learning on a Web browser according to the user authority as shown in Fig.2 (2-1.Learning Management Interface). User can easily load his own data on DB throughout this resource management scheme, and we can classify and deal with data based on the stuck of resources.

User can create new works using data uploaded through resource management, and can visually confirm the procedure of this learning realizations as shown in Fig.2 (2-2.Intuitive Learning Fulfillment). In the present module, learning process consists of the following 4 sequential steps; The 1st step is preprocess of data and divide the learning data according to the number given by user so that the tasks can be performed in a multi-processing scheme as the 2nd step. At the 3rd step, user selects and integrates the data. Finally data learning realization will be finished as the 4th step.

User can reuse the result from each step mentioned above combining new tasks as show in Fig.2 (2-3.Reuse of Resource and Results). Throughout these task and resource management processes, we can share not only user's own data but data shared by other users so that we can realize an efficient scheme which fulfills only novel tasks without any repetition of the

same tasks done in the past. Therefore it is possible for user to learn only newly added data incrementally without any relearning of total stuck. Skills for data reuse are very useful in this Big Data environment where a lot of huge data bundles are continuously generated.

Finally user can test the module performance of learning results by pressing the 'test' button as shown in Fig.2 (2-4.Result Analysis). After finishing all the learning processes, we can confirm the performance of learning objects and their results as a visual aid. Test results consist of the followings; Calculation outputs of performance index, Visual graphs of performance evaluation, Analytics statistical information, and their saving and management features independently.

4. PERFORMANCE EVALUATION

Classification algorithms are currently difficult to apply to the environment of Big Data where lots of data are continuously generated because they perform classification tasks based on memory. In this study, we implement all the input and output processes based on DB so that they can be efficiently used in the Big Data environment by expanding the module of traditional Feature Voting Classification algorithm. We also solve the problem of memory burden using DB on which we load all the original data we should classify and their input/output and all the results of operations in classification process.

For this performance evaluation, we realize a program module which loads the results on database after parsing the original data. The process of data mining module deals with the data input/output through DB adapter. We can proceed our tasks as a multi-process or distributed parallel process environment because the original data have been loaded on DB and divided into segments of learning to be processed as parallel alignments. We implement the module of calculation for the values such as precision, recall, accuracy, f-measure, and macro/micro averaged values as their class level. We can evaluate the performance using the results from classification tasks following the learning steps.

4.1 Experimental Data Set

We collected about a million articles classified as 8 categories like IT/Science, Economics, Society, Life/Culture, World, Sports, Entertainments, and Politics from the Naver News for our experiment of non-structured data. In addition we assign learning data and verification data as 80% and 20% respectively. This ratio is evenly applied to 8 categories mentioned above. Table2 shows the statistics of data set collected.

Table2. Distribution of Non-structured Data

Class	# of Document	# of Training set	# of Test set
IT/Science	125,784	100,629	25,155
Economics	125.775	100,620	25,155
Society	125,781	100,624	25,157
Life/Culture	125,797	100,636	25,161
World	125.651	100,516	25,135
Sports	125,780	100,620	25,160
Entertainment	107,643	86,118	21,525
Politics	125,819	100,653	25,166
Total	**988,030**	**790,416**	**196,614**

4.2 Experiment Method

Performance evaluation is fulfilled to verify the incremental learning scheme and its parallel and integrated processing as an experimental aid. As shown in Fig.3, we divided the experimental learning data into 12 segments and perform learning procedure, which is followed by a sequential integration of results expanding the stuck of these results of tasks mentioned above. We can verify that it is possible to learn large amount of data without repetition of relearning process under the Big Data environment. Also we prove that our method has better performance than traditional learning schemes.

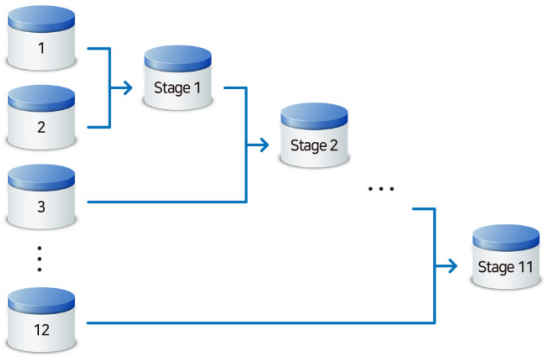
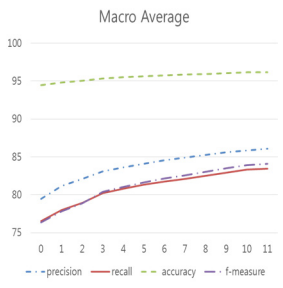


Fig. 3 Incremental Process of Learning Data Set

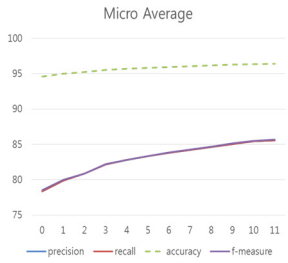
4.3. Performance Measurement

Increment of each step of the incremental learning is about 65,000 articles of learning data when we are evaluating performance. Fig. 4 and 5 show the results of Macro Averaged and Micro Averaged performance tests respectively. Both graphs reveal averages according to the size of data or 0~11 points. We are able to confirm that Macro/Micro Avg. performance index are gradually enhanced as the size of learning data are increased as shown in both graphs.



point	precision	recall	accuracy	f-measure
point 0	79.48	76.54	94.45	76.34
point 1	81.17	78.01	94.82	77.85
point 2	82.08	78.96	95.06	78.89
point 3	83.09	80.22	95.37	80.39
point 4	83.64	80.81	95.52	81.03
point 5	84.11	81.32	95.65	81.63
point 6	84.55	81.75	95.75	82.15
point 7	84.90	82.13	95.85	82.59
point 8	85.27	82.54	95.95	83.07
point 9	85.64	82.94	96.05	83.53
point 10	85.84	83.31	96.14	83.94
point 11	86.10	83.47	96.19	84.11

Fig. 4 Macro Averaged Performance Evaluation



point	precision	recall	accuracy	f-measure
point 0	78.37	78.37	94.59	78.53
point 1	79.89	79.89	94.97	79.97
point 2	80.87	80.87	95.22	80.90
point 3	82.18	82.18	95.55	82.23
point 4	82.80	82.80	95.70	82.81
point 5	83.33	83.33	95.83	83.35
point 6	83.79	83.79	95.95	83.85
point 7	84.19	84.19	96.05	84.26
point 8	84.61	84.61	96.15	84.71
point 9	85.03	85.03	96.26	85.13
point 10	85.42	85.42	96.35	85.50
point 11	85.58	85.58	96.40	85.65

Fig. 5 Micro Averaged Performance Evaluation

5. EXCELLENCE OF THE ACHIEVEMENT

Recently there is a dominant trend that current data processing technologies are advanced toward Big Data aiming direction as demands of analyses and applications of Big Data are abundantly increased. Classification technologies, however, have a critical limitation in velocity of creation and variety of data structure which are more important characteristics rather than storage and retrieval of large amount of data. Traditional classification technologies, nonetheless, still focus on memory, search, and extract of Big Data.

The classification technology based on incremental learning concepts in this paper is developed to solve these critical hazards of the Big Data environment, which contributes to submit an only appropriate method to classify structured and unstructured data simultaneously.

5.1 Structured and Unstructured Data Processing

First of all, we advanced classification technology of semi-structured and/or unstructured data, which is one of the most important topics in Big Data analysis. We also have submitted performance evaluation results. On the other hand, we developed a processing module for structured data so that we can deal with various type of Big Data and advance their classification in a wider range. Experiment of comparison shows that our incremental learning scheme get along with traditional classifier like NB or SVM in performance evaluation, and moreover it supports gradual process of learning without re-learning objects so that we can obtain a better real-time processing capability.

5.2 Reusability of Data through Incremental Learning

Incremental learning method is more efficient in classifications of Big Data environment because it can reduce repetitive re-learning objects. Also its classification performance is due to enhanced according to the data accumulated continuously. User, therefore, easily deal with time-series data and analyze large amount of data under the condition of data re-using by this incremental learning scheme. In addition, we can easily expand its application area because its learning results can be used or shared with any other purpose.

5.3 Development of a Convenient User Interface

We developed a Web interface to let users easily approach to the classifier without any expert knowledge. The interface is commonly explicit and easy to use so that we can re-use learning data already analyzed and manage our learning tasks through this Web interface. We can share the metadata of experiments and their results between users in order to perform data processing for huge amount of Big Data efficiently. Large bundle of data are automatically divided into learning segments to be learned by the classifier and so forth integration process would be fulfilled. Series of processes are possible to be proceeded by 2 or 3 clicking the buttons in the interface. From the data input to the result of performance evaluation, all the process consist of automated simple interfaces.

5.4 Big Data Mining by Distributed Parallel Framework

Recently technology development focuses on real-time data processing and how to use these data according to the rapid environment change of various business areas. Struggles for reducing response time to the real-time level throughout clustering and parallel processing are globally devoted in this mining world.

Algorithm for incremental learning is appropriate to the SN(shared-nothing) structure among distributed processing architectures because it performs data tasks on DB base not on memory base. We can make use of this incremental learning concept as a data classifier by building an independent module and loading on the data analysis layer of the Big Data eco system.

5.5 Micro Accuracy Performance of Incremental Learning

Accuracy of each classifier shows about 95% like in Fig.6. Comparing our algorithm with ones of NB(Naïve Bayes) and SVM(support vector machine) which are commonly used methods, we can get accuracy of almost the dame in all the three models.

Stage	fvc	svm-c-1000	svm-c-500	bayes
1	94.59	95.77	95.65	94.98
2	94.97	96.16	95.99	95.41
3	95.22	96.36	96.18	95.61
4	95.55	96.44	96.26	95.73
5	95.70	96.56	96.36	95.83
6	95.83	96.64	96.45	95.89
7	95.95	96.72	96.48	95.96
8	96.05	96.76	96.51	96.02
9	96.15	96.79	96.55	96.05
10	96.26	96.83	96.55	96.09
11	96.35	96.85	96.62	96.10
12	96.40	96.87	96.64	96.12

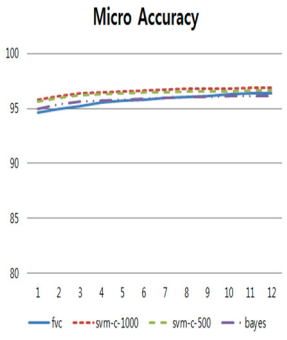


Fig. 6 Accuracy Comparison

Therefore we can conclude that our incremental learning algorithm has a realistic advantage in the process of studying only new data without any re-learning of old data, on the other hand it also has a meaningless difference in accuracy performance.

6. CONCLUSION

Traditional learning mechanism usually analyzes total data including ones already learned and stuck at every changing or

adding, on the other hand, incremental learning module analyzes only newly added data without any re-learning old stuck of data. Applying this technology, user can bring an old data block through resource management scheme and merge on a new data bundle in any task processing. Efficiency of this incremental learning can advance the speed of data creation which is important to overcome the hazard of Big Data processing world.

Using this incremental learning model of commercial version, we can classify unstructured data as well as various structured data from variety of sensors and IoT technology items. We expect almost all types of data can be effectively processed using this incremental learning in order to make a synergy effect on the whole industry. As a concluding remark, our method will be a good substitute of high performance large computing system for Big Data analysis including large-scale sparse matrix using a PC cluster environment. We also expect more about things that high performance computing environment reveals a lot much higher efficiency with the scheme proposed in this paper

ACKNOWLEDGEMENTS

This research was partially supported by the sabbatical program of Mokwon University, from Sep.2018 to Feb.2019.

REFERENCES

[1] Jang-Won Gim, Myung-Gwon Hwang, Sa-Kwang Song, Jin-Hyung Kim, Do-Heon Jeong, and Han-Min Jung, “Researcher history tracking service for prescriptive analytics based on researcher activities,” Journal of KIISE: Computing Practices and Letters, vol. 20, no. 6, 2014, pp. 359-363.

[2] Do-Heon Jeong, “A study on automatic database selection technique using the maximal concept strength recognition method,” Journal of the Korean Society for Information Management, vol. 27, no. 3, 2010, pp. 265-281. https://doi.org/10.3743/kosim.2010.27.3.265

[3] Do-Heon Jeong, Hwan-Min Kim, Hye-Sun Kim, and Ki-Jeong Shin, “The relationship between the specificity of S&T terms and auto-classification accuracy,” Proceedings of the 14th Conference of Korean Society for Information Management, 2007, pp. 31-36.

[4] Jae-Yun Lee, “Improving the performance of a fast text classifier with document-side feature selection,” Journal of Information Management, vol. 36, no. 4, 2005, pp. 51-69. https://doi.org/10.1633/jim.2005.36.4.051

[5] Jae-Yun Lee, “A novel clustering method for examining and analyzing the intellectual structure of a scholarly field,” Journal of the Korean Society for Information Management, vol. 23, no.4, 2006, pp. 215-231. https://doi.org/10.3743/kosim.2006.23.4.215

[6] Won-Goo Lee, Sung-Ho Shin, Kwang-Young Kim, Do-Heon Jeong, Hwa-Mook Yoon, Won-Kyung Sung, and Min-Ho Lee, “Semi-automatic management of classification scheme with interoperability,” The Journal of the Korea Contents Association, vol. 11, no. 12, 2011, pp. 466-474. https://doi.org/10.5392/jkca.2011.11.12.466

[7] R. Burke, “Hybrid Recommender Systems : Survey and Experiments,” User Modeling and User-Adapted Interaction, vol. 12, no. 4, Nov. 2011, pp. 331-370.

[8] Francesco Ricci, *Recommender Systems Handbook*, Springer, 2011.

[9] G. Adomavicius and A. Tuzhilin, “Toward the Next Generation of Recommender Systems : A Survey of the State-of-the-Art and Possible Extensions,” IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, 2005, pp734-749.

[10] R. Plutchik, “The Nature of Emotions,” American Scientist, vol. 89, no. 4, 2001, pp. 344-350.

[11] J. A. Russell, J. M. Fernabdez-Dols, A. S. R. Mastead, and J. C. Wellenkamp, *Everyday Conceptions of Emotion : An introduction to the Psychology, Anthropology and Linguistics of Emotion*, Kluwer Academic Publishers, 1995.

[12] Jeong-Won Lee, Byung-Won Min, and Yong-Sun Oh, “Design of Moa Contents Curation Service System Based on Incremental Learning Technology,” Proceedings of the Korea Contents Association 2018 Spring Symposium, 2018, pp.401-402.

[13] Jeong-Won Lee, Byung-Won Min, and Yong-Sun Oh, “Design and Implementation of Contents Curation LOD(Linked Open Data) Cloud Service System Based on Incremental Learning Model for Big Data Mining,” Proceedings of the Korea Contents Association 2018 International Conference on Convergence Content, 2018, pp.131-132.

[14] Won-Goo Lee, Myung-Gwon Hwang, Min-Ho Lee, Sung-Ho Shim, Kwang-Young Kim, Hwa-Mook Yoon, Won-Kyung Sung, and Do-Heon Jeon, “The Automatic Management of Classification Scheme with Interoperability on Heterogeneous Data,” Journal of the Korea Institute of Information and Communication Engineering, vol. 15, no. 12, 2011, pp. 2609-2618.

Byung-Won Min

He received M.S. degree in computer software from Chungang University, Seoul, Korea in 2005. He worked as a professor in the dept. of computer engineering, Youngdong University, Chungbuk, Korea, from 2005 to 2008. He received Ph.D. degree in the dept. of Information and Communication Engineering, Mokwon University, Daejeon, Korea, in 2010. He is currently a professor of Mokwon University since 2010. His research interests include digital communication systems, information theory and their applications. He is recently interested in multimedia content and Big Data.

Yong-Sun Oh

He received B.S., M.S., and Ph.D. degrees in electronic engineering from Yonsei University, Seoul, Korea, in 1983, 1985, and 1992, respectively. He worked as an R&D engineer at System Development Division of Samsung Electronics Co. Ltd., Kiheung, Kyungki-Do, Korea, from 1984 to 1986. He joined as a faculty of Division of Information and Communication Convergence Engineering, Mokwon University, Daejeon, Korea, in 1988. During 1998-1999 he worked as a visiting professor of KMU, Busan, Korea, where he was nominated as Head of Academic Committee of KIMICS. Afterward, he returned to Mokwon University and served as Dean of Central Library Information Center for 2 years and as Director of Corporation of Industrial Educational Programs for 2 years. Recently he also served as Dean of Engineering College, Dean of Management Strategy, and Dean of Academic Affairs, Mokwon University. During his sabbatical years, he worked as an Invited Researcher at ETRI in 2007~2008, at KISTI in 2014~2015. He had been serving as the President of KoCon from 2006 to 2012. He is currently a president emeritus of KoCon and a professor of Mokwon University. His research interests include communication systems, information theory, and their applications. Recently he is interested in multimedia content and personalized e-Learning.