Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A DYNAMIC DATA FUSION METHOD AND ITS APPLICATION

[1] RONGCHUN WU, [2] FENGLI ZHANG

[1]University of Electronic Science and Technology School of Computer Science, Police Academy, Chengdu 610213, Sichuan, China

[2] University of Electronic Science and Technology School of Computer Science, Chengdu 610054, Sichuan, China

ABSTRACT

Data fusion involves multi-sources or multi-presentations of a single source to perform inferences which are more comprehensive and accurate than those of any single method. Thus, data fusion makes it possible to create a synergistic process in which the consolidation of individual data creates a combined resource with a productive value greater than the sum of its parts. While considerable research has been done on data fusion in the past, most of them performed in the field of multi-sensor fusion. There has been relatively less work conducted in a data mining context. As the form factor of computing and communicating devices shrinks and the capabilities of such devices continue to grow, it has become reasonable to imagine applications that require rich computing resources today becoming viable candidates for future sensor networks.

Keywords: *Data Fusion, Dynamic Data, Sensor Networks*

1. INTRODUCTION

There are two main branches in data modeling, descriptive modeling and predictive modeling. Descriptive modeling is also called exploratory data analysis (EDA). The purpose of descriptive modeling is to extract more compact and concise information from a large amount of data sources to get insight on patterns in these data. The technologies used by descriptive modeling span from the simplest statistical measures of variables, e.g. mean, variance, skewness and correlation coefficient, to more complex models, e.g. clustering, probability density estimation and dimensionality reduction. In other words, descriptive modeling is not defined by a set of techniques, but rather by the need to account for the implied structures in the data in a compact way with better interpretability.

Unlike descriptive modeling which identically treats all variables, predictive modeling separates the variables into two groups, predictors (the independent variable) and response (the dependent variables). The purpose of predictive modeling is to find strong links between predictors and response which can be used to predict the new observations without response measures Data fusion is an emerging technique that attempts to improve the precision and correctness of any one method which is limited by its specific inherent disadvantages. Data fusion is rapidly emerging from ever increasing military programs and has been extended into more broad non-military areas, such as the academic, commercial and industrial communities. In general, data fusion involves multi-sources or multi-presentations of a single source to perform inferences which are more comprehensive and accurate than that of any single method. Thus, data fusion makes it possible to create a synergistic process in which the consolidation of individual data creates a combined resource with a productive value greater than the sum of its parts [1]. The first reports of the automation of data fusion functions are from the late 1970s. Throughout the 1980s, a lot of research was done with regard to data fusion [2-6]. This research was mainly conducted by the three U.S. military services, and much of its results were published in open literature. To improve the common understanding and communication among researchers in the field of data fusion, the U.S. Joint Directors of Laboratories (JDL) Data Fusion Working Group (DFS), established in 1986, began to define the terminology related to data fusion. In the late 1980s a small number of military data fusion systems were operational. Since then, data fusion technology has rapidly advanced. What started as a loose collection of related technologies became an emerging engineering discipline[7-9]. By the end of the 1980's, two national conferences on data fusion were conducted annually, including a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

conference sponsored by the US DoD Joint Directors of Laboratories (JDL) - Technology Panel for C3, and a second conference sponsored by the International Society of Optical Engineering (SPIE). Although most of the research projects are successfully sponsored by DoD in military surveillance and land-based battlefield management systems[10], the applications of data fusion are also growing rapidly in commercial endeavors (e.g. robotics, intelligent building[11] and medical image processing), non-military government projects (e.g. environment surveillance[12-15], intelligent transportation system[16], weather forecasting [17,18]) and industrial projects (e.g. condition-based maintenance, industrial process control). A more recent idea is the application of multi-sensor data fusion techniques to the area of information security[19]. Data fusion methods have been extensively used in applications where multiple sources of data are widely available. From a statistical point of view, each measurement used for multidimensional and multivariate analysis will reduce the varying amount of uncertainty or variance of the interested target. With the redundant information of the multi-source data, we can improve the reliability of inference; with the complementary information, we can improve the capability of inference. Although the JDL model and its variations have been commonly applied to various different applications, there is no common one-fits-all architecture for these systems because of their diverse nature. The challenge of a data fusion system is to determine how to combine varying quality data in terms of value for modeling to generate reliable results which achieve some expected accuracy. If not done properly, one set of the data with bad quality may worsen the predictive power of the existing model. In a particular situation, one subset of available measurements could be the only good source for fusion. Recent data fusion research has addressed time series and image based data analysis involving the target tracking, characterization, and identification of dynamic entities, but only a few publications concern predictive modeling systematically in a data mining context. Putten[20] employed data fusion through statistical matching for internal and external evaluation during customer data analysis. More statistic matching references can be found in [5,21-22]. The research on data fusion for predictive modeling in data mining is still a loose collection of related technologies as mentioned above. So building a general functionally-oriented model and architecture for predictive data mining is quite useful to provide a clear overview of the taxonomy of the associated technologies. The purpose of this thesis is to build an auto-fusion framework and to establish procedures and guidelines for data fusion in predictive data mining with multiple commensurate data sources available, i.e. all datasets are generated from the same collection of objects and have the same number of observations. In addition to the framework, advanced algorithms like K-PLS based ensemble and the kernel fusion method are developed and joined the associated techniques with particular architecture to help get more accurate and robust models while building a fusion system for various applications.

2. APPLICATION CONTEXT AND REQUIREMENTS

A fusion application has the following characteristics: (1) it is continuous in nature,(2) It requires efficient transport of data from/to distributed sources/sinks, and (3) it requires efficient in-network processing of application-specified fusion functions. A data source may be a sensor (e.g., camera) or a standalone program; a data sink represents an end consumer and includes a human in the loop, an actuator (e.g., a fire alarm), an application (e.g., a data logger), or an output device such as a display; a fusion function transform the data streams (including aggregation of separate streams into a composite one) en route to the sinks. Thus a fusion application is a directed task graph: the vertices are the fusion functions, and the edges represent the data flow (i.e., producer-consumer relationships) among the fusion points (cycles—if any—represent feedback in the task graph).This formulation of the fusion application has a nice generality. It may be an application in its own right (e.g., video based surveillance). It allows hierarchically composing a bigger application (e.g., an emergency response) wherein each component may itself be a fusion application (e.g., image processing of videos from traffic cameras). It allows query processing by overlaying a specific query (e.g., “show a composite video of all the traffic at the spaghetti junction”) on to the task graph. Consider, for example, a video-based surveillance application. Cameras are deployed in a distributed fashion; the images from the cameras are filtered in some application-specific manner, and fused together in a form that makes it easy for an end user (human or some program) to monitor the area. The compute-intensive part may analyze multiple camera feeds from a region to extract higher-level information such as “motion,” “presence or absence

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of a human face," or "presence or absence of any kind of suspicious activity." The fusion functions may result in contraction or expansion of data flows in the network. For example, the filter function selects images with some interesting properties (e.g., a rapidly changing scene), and sends the compressed image data to the collage function. Thus the filter function is an example of a fusion point that does data contraction. The collage function uncompressed the images coming from possibly different locations. It combines these images and sends the composite image to the root (sink) for further processing. Thus, the collage function represents a fusion point that may do data expansion. Given the pace of technology, it is conceivable to imagine future sensor networks wherein some nodes have the computational capability of today's handhelds (such as an iPAQ), and communication capabilities equivalent to Bluetooth, 802.11a/b/g, 802.15.3 (WPAN), or even UWB (up to 1 Gb/s). While a quest for smaller footprint devices with lower cost continues, we expect that there will have a continuum of capabilities from the Berkeley motes to today's handhelds. Recent advances in low-power microcontrollers, and increased power-conscious radio technologies lend credence to this belief. For example, next-generation iMote prototypes (go onlineto http://www.intel.com/research/exploratory/motes.htm) and Telos motes [Polastre et al. 2004] are available for research now. Although not as computationally powerful as a modern iPAQs, iMotes provide 12-MHz 32-bit ARM7TDMI processors and 64-kB RAM/512-kB FLASH, a significant increase in capability compared to Berkeley mote MICA2 (go online to http://www.xbow.com/Products/productsdetails.aspx?sid=72) predecessors that only had 8MHz 8-bit ATmega128L microcontrollers with 4-kB RAM/128-kB FLASH. Furthermore, the wireless bandwidth available with iMotes is Bluetooth based (over 600-Kb/s application-level bandwidth), greatly exceeding Berkeley motes'–38.4-Kb/s data rate. Coupled with this trend, high-bandwidth sensors such as cameras are becoming ubiquitous, cheaper, and lighter (in this case possibly due to the large-scale demands of cell-phone manufacturers for these cameras, where camera phone shipment is expected to reach 903 million in 20101). Thus we envision future wireless sensor networks deployments to consist of high bandwidth and powerful sensor/actuator sources and infrastructures coexisting with more constrained nodes, with energy still being a scarce resource.

3. DYNAMIC DATA FUSION METHOD

3.1 Architectural

We have designed the Data Fusion architecture to cater to the evolving application needs and emerging technology trends. We make some basic assumptions about the execution environment in the design of Data Fusion.

The application level input to the architecture are (1) an application task graph consisting of the data flows and relationship among the fusion functions, (2) the code for the fusion functions (currently supported as C program binaries), (3) a cost function that formalizes some application quality metric for the sensor network (e.g., "keep the average node energy in the network the same"). The task graph has to be mapped over a large geographical area. In the ensuing overlay of the task graph on to the real network, some nodes may serve as relays while others may perform the application-specified fusion operations. The fusion functions may be placed anywhere in the sensor network as long as the cost function is satisfied. All source nodes are reachable from the sink nodes. Every node has a routing layer that allows each node to determine the route to any other node in the network. This is in sharp contrast to most current day sensor networks that support all-to-sink style routing. However, the size of the routing table in every node is only proportional to the size of the application task graph (to facilitate any network node in the ensuing overlay to communicate with other nodes hosting fusion functions) and not the physical size of the network.

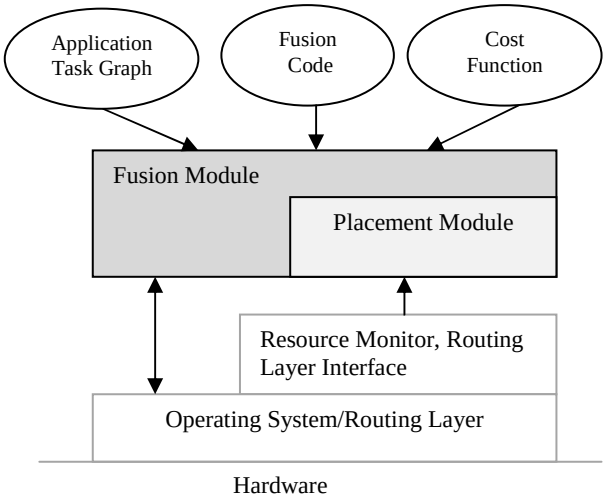

Figure 1: Data Fusion Architecture

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2 Fusion Module

The fusion module consists of the shaded components shown in Figure 2. It is implemented in C as a layer on top of the Stampede runtime system. All the buffers (input buffers, fusion buffer, and prefect buffer) are implemented as Stampede channels. Since Stampede channels hold time stamped items, it is a straightforward mapping of the fusion attribute to the timestamp associated with a channel item. The Status and Command registers of the fusion architecture are implemented using the Stampede register abstraction. In addition to these Stampede channels and registers that have a direct relationship to the elements of the fusion architecture, the implementation uses additional Stampede channels and threads. For instance, there are prefetch threads that gather items from the input buffers, fuse them, and place them in the prefetch buffer for potential future requests. This feature allows latency hiding but comes at the cost of potentially wasted network bandwidth and hence energy (if the fused item is never used). Although this feature can be turned off, we leave it on in our evaluation and ensure that no such wasteful communication occurs. Similarly, there is a Stampede channel that stores request that is currently being processed by the fusion architecture to eliminate duplication of work. The create FC call from an application thread results in the creation of all the above Stampede abstractions in the address space where the creating thread resides. An application can create any number of fusion channels (modulo system limits) in any of the nodes of the distributed system. An attachFC call from an application thread results in the application thread being connected to the specified fusion channel for getting fused data items. For efficient implementation of the getFCItem call, a pool of worker threads is created in each node of the distributed system at application startup. These worker threads are used to satisfy getFCItem requests for fusion channels created at this node. Since data may have to be fetched from a number of input buffers to satisfy the getFCItem request, one worker thread is assigned to each input buffer to increase the parallelism for fetching the data items. Once fetching is complete, the worker thread rejoins the pool of free threads. The worker thread to fetch the last of the requisite input items invokes the fusion function and puts the resulting fused item in the fusion buffer. This implementation is performance-conscious in two ways: first, there is no duplication of fusion work for the same fused item from multiple requesters; second, fusion work itself is parallelized at each node through the worker threads.

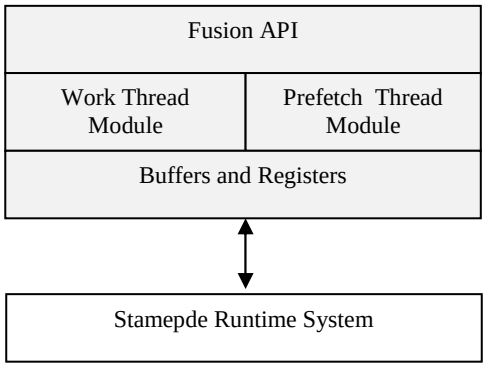

Figure 2: Fusion Module Components

4. NUMERICAL EXAMPLES

In this section, we provide several numerical results to help understand the coverage performance under the data fusion model. We adopt the signal decay function with k = 2. Fig. 3 plots the approximate coverage computed. We can see from Fig. 3 that the coverage initially increases with fusion range R, but decreases to zero eventually. Intuitively, as the fusion range increases, more sensors contribute to the data fusion resulting in better sensing quality. However, as R becomes very large, the aggregate noise starts to cancel out the benefit because the target signal decreases quickly with the distance from the target. In other words, the measurements of sensors far away from the target contain low quality information and hence fusing them leads to lower detection performance. An important question is thus how to choose the optimal fusion range (denoted by Ropt) that maximizes the coverage. First, the Ropt can be obtained through numerical experiments. Fig. 4 plots the optimal fusion ranges under different network densities

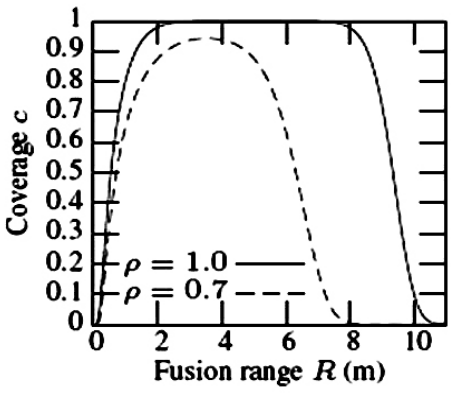

Figure 3: Coverage

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

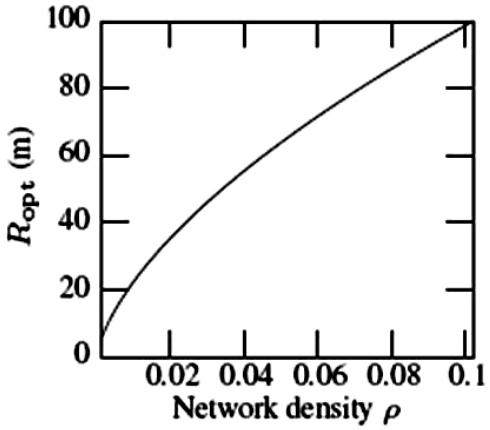

Figure 4: Optimal Fusion Range

5. CONCLUSION

Sensing coverage is an important performance requirement of many critical sensor network applications. In this paper, we explore the fundamental limits of coverage based on stochastic data fusion models that jointly process noisy measurements of sensors. The scaling laws between coverage, network density, and signal-to-noise ratio (SNR) are derived. Data fusion is shown to significantly improve sensing coverage by exploiting the collaboration among sensors. Our results help understand the limitations of the existing analytical results based on the disc model and provide key insights into the design and analysis of WSNs that adopt dat fusion algorithms.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation (Grant No. 61133016), and the National High Technology Joint Research Program of China (863 Program, Grant No. 2011AA010706)

REFERENCES:

[1] Y. Tamura and S. Yamada, "Validation of an OSS reliability assessment method based on ANP and SRGM's," P*roceedings of the International Workshop on Recent Advances in Stochastic Operations Research, Canmore*, Canada, August 25–26, 2005, pp. 273–280.

[2] Lin DongMei, Liu Jun, "Research on Train Timetable-based Railway Route Planning Problem", *AISS: Advances in Information Sciences and Service Sciences*, Vol. 4, 2012, No. 14, pp. 71-79.

[3] Shang Gao, Cungen Cao, "Convergence Analysis of Particle Swarm Optimization Algorithm", *AISS: Advances in Information Sciences and Service Sciences*, Vol. 4, 2012, No. 14, pp. 25-32.

[4] D. Hall and J. Llinas, "An introduction to multi-sensor data fusion," *Proceedings of the IEEE*, Vol. 85, No. 1, 1997, pp. 6 –23,

[5] N. Ahmed, S. Kanhere, and S. Jha, "Probabilistic coverage in wireless sensor networks", In LCN, 2005, pp. 22-26.

[6] P. Pohanka, J. Hrabovsky, and M. Fiedler, "Sensors simulation environment for sensor data fusion", *Proceedings of the 14th International Conference on in Information Fusion (FUSION)*, 2011, pp. 1–8,

[7] J. Sabater, C. Serria, "Review on computational trust and reputation models", *Artificial Intelligence Review*, Vol. 24, No. 1, 2005, pp. 33-60.

[8] N. Bisnik, A. Abouzeid, "Stochastic eventcapture using mobile sensors subject to a quality metric", *Proceedings of MobiCom 2006*, March 16-19, 2006, pp.14-17.

[9] Q. Shen, R. Leitch, "Fuzzy Qualitative Simulation", IEEE Trans, on Systems, Man, and Cybernetics, Vol. 23, No. 4, 2011, pp. 17-19.

[10] P. Brass, "Bounds on coverage and target detection capabilities for models of networks of mobile sensors", *ACM Trans. Sen. Netw*, Vol. 3, No. 2, 2007, pp. 1-12.

[11] Z. Chair, P. Varshney, "Optimal data fusion in multiple sensor detection systems", *IEEE Trans, Aerosp, Electron, Syst*, Vol. 22, No. 1, 1990, 12-16.

[12] T. Clouqueur, K. K. Saluja, "Fault tolerance in collaborative sensor networks for target detection", *IEEE Trans. Comput*, Vol. 53, No. 3, 2004, pp. 43-45.

[13] W.P. Chen, L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks", *IEEE Trans. Mobile Comput*, Vol. 3, No. 3, 2004, pp. 21-24.

[14] S. Y. Cheung, S. Coleri, B. Dundar, S. Ganesh, C. W. Tan, P. Varaiya. A sensor network for traffic monitoring (plenary talk), *Proceedings of IPSN*, 2004.

[15] T. Clouqueur, K. Saluja, "Fault tolerance in collaborative sensor networks for target detection", *IEEE Trans. Comput.*, Vol. 53, No. 3, 2004, pp. 63-69.

[16] R. S. Sandhu, D. Ferraiolo, R. Kuhn, "The NIST model for Role-based access control:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

towards a unified standard", *Proceedings of the fifth ACM workshop on Role-based access control*, pp. 47-63, 2000.

[17] X. Gu, K. Nahrstedt, "On composing stream applications in peer-to-peer environments", *IEEE Trans. Parball. Distrib. Syst.*, 2006, pp.824–837.

[18] J. Hwan, C. Leandros, "Energy conserving routing in wireless ad-hoc networks", *Proceedings of IEEE Infocom*, February 22–31, 2008, pp. 126-130.

[19] S. A. Ludwig, V. Pulimi, A. Hnativ, "Fuzzy Approach for the Evaluation of Trust and Reputation of Services", *Proceedings of the 18th international conference on Fuzzy Systems*, USA, 2009, 115-120.

[20] E .Cayirci, W. Sankarasubramanian, "Wireless sensor networks: A survey", *Comput. Netw*., Vol. 38, No. 4, 2002, pp. 393–422.

[21] J. Hill, et al., "System architecture directions for networked sensors", *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, 2000, pp. 93– 104.

[22] Johnson, Thomas H, "Mission Space Model Development, Reuse and Conceptual Models of the Mission Space Toolset", *Spring Simulation Interoperability Workshop Thesis's*, March 2009, Vol. 2, pp. 893-900.