Signal Processing, Sensor/Information Fusion, and Target Recognition XXVIII

Ivan Kadar
Erik P. Blasch
Lynne L. Grewe
Editors

15–17 April 2019
Baltimore, Maryland, United States

Sponsored and Published by
SPIE

Volume 11018
## Contents

- Authors vii
- Conference Committee ix
- Introduction to the Invited Panel Discussion xiii
- Invited Panel Slides xv

### SESSION 1  MULTISENSOR FUSION, MULTITARGET TRACKING AND RESOURCE MANAGEMENT I

<table>
<thead>
<tr>
<th>Paper ID</th>
<th>Title</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101802</td>
<td>Extracting fast targets from an EO sensor</td>
<td>[11018-1]</td>
</tr>
<tr>
<td>1101803</td>
<td>Target tracking in over the horizon radar</td>
<td>[11018-2]</td>
</tr>
<tr>
<td>1101804</td>
<td>On-demand track-to-track fusion using local IMM inside information</td>
<td>[11018-3]</td>
</tr>
<tr>
<td>1101805</td>
<td>CRLB for multi-sensor rotational bias estimation for passive sensors without target state estimation</td>
<td>[11018-4]</td>
</tr>
<tr>
<td>1101806</td>
<td>The cross-covariance for heterogeneous track-to-track fusion</td>
<td>[11018-5]</td>
</tr>
</tbody>
</table>

### SESSION 2  MULTISENSOR FUSION, MULTITARGET TRACKING AND RESOURCE MANAGEMENT II

<table>
<thead>
<tr>
<th>Paper ID</th>
<th>Title</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101808</td>
<td>Experimental results in bearings-only tracking using the sequential Monte-Carlo probability hypothesis density filter</td>
<td>[11018-7]</td>
</tr>
<tr>
<td>1101809</td>
<td>&quot;Self-intoxication&quot; and the effects on track quality</td>
<td>[11018-9]</td>
</tr>
<tr>
<td>110180A</td>
<td>Problems with information fusion under extreme multiple-counting conditions</td>
<td>[11018-8]</td>
</tr>
<tr>
<td>110180B</td>
<td>A fuzzy inference system for ship-ship collision alert generation</td>
<td>[11018-10]</td>
</tr>
<tr>
<td>110180C</td>
<td>Maritime situational awareness with OCULUS Sea C2i and forensics tools for a Common Information Sharing Environment (CISE)</td>
<td>[11018-100]</td>
</tr>
<tr>
<td>SESSION 3</td>
<td>INFORMATION FUSION METHODOLOGIES AND APPLICATIONS I</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>110180D</td>
<td>A GLMB filter for unified multitarget multisensor management [11018-11]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SESSION 4</th>
<th>INFORMATION FUSION METHODOLOGIES AND APPLICATIONS II</th>
</tr>
</thead>
<tbody>
<tr>
<td>110180G</td>
<td>Sequential and parallel fusion of detection and classification systems [11018-14]</td>
</tr>
<tr>
<td>110180H</td>
<td>Sequence theory for classification in multi-label ATR classification tasks [11018-15]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SESSION 5</th>
<th>INFORMATION FUSION METHODOLOGIES AND APPLICATIONS III</th>
</tr>
</thead>
<tbody>
<tr>
<td>110180I</td>
<td>Extremely deep Bayesian learning with Gromov’s method [11018-16]</td>
</tr>
<tr>
<td>110180K</td>
<td>Information fusion to estimate resilience of dense urban neighborhoods [11018-18]</td>
</tr>
<tr>
<td>110180L</td>
<td>An agent-administrator-based security mechanism for distributed sensors and drones for smart grid monitoring [11018-19]</td>
</tr>
<tr>
<td>110180M</td>
<td>Object recognition, identification and classification for intelligent surveillance and reconnaissance platforms [11018-20]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SESSION 6</th>
<th>INFORMATION FUSION METHODOLOGIES AND APPLICATIONS IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>110180N</td>
<td>Long lasting effects of awareness training methods on reducing overall cyber security risk [11018-21]</td>
</tr>
<tr>
<td>110180O</td>
<td>Automated real-time risk assessment for airport passengers using a deep learning architecture [11018-22]</td>
</tr>
<tr>
<td>110180P</td>
<td>A framework for context change detection and management in probabilistic models for context in fusion [11018-23]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SESSION 7</th>
<th>INFORMATION FUSION METHODOLOGIES AND APPLICATIONS V</th>
</tr>
</thead>
<tbody>
<tr>
<td>110180Q</td>
<td>Deep learning in AI and information fusion panel discussion [11018-24]</td>
</tr>
<tr>
<td>110180S</td>
<td>Applying cognitive psychology principles to the (dis)information environment: an examination of discourse comprehension, memory, and fusion of news articles [11018-26]</td>
</tr>
<tr>
<td>110180T</td>
<td>Evaluation of algorithms for fake news identification [11018-27]</td>
</tr>
<tr>
<td>Session 8</td>
<td>Signal and Image Processing, and Information Fusion Applications I</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>110180V</td>
<td>Archaeological dating using a data fusion approach [11018-29]</td>
</tr>
<tr>
<td>110180W</td>
<td>ULearn: understanding and reacting to student frustration using deep learning, mobile vision and NLP [11018-30]</td>
</tr>
<tr>
<td>110180X</td>
<td>Drone based user and heading detection using deep learning and stereo vision [11018-31]</td>
</tr>
<tr>
<td>110180Y</td>
<td>Deep learning on hyperspectral data to obtain water properties and bottom depths [11018-32]</td>
</tr>
<tr>
<td>110180Z</td>
<td>Characterizing hyperspectral signatures of human faces in the shortwave infrared spectrum [11018-33]</td>
</tr>
<tr>
<td>1101810</td>
<td>Comparison of neural network classifiers for automatic target recognition [11018-34]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Session 9</th>
<th>Signal and Image Processing, and Information Fusion Applications II</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101812</td>
<td>Deep learning architecture advancements for accurate and robust image registration [11018-36]</td>
</tr>
<tr>
<td>1101813</td>
<td>Radar target identification using HRRP-based features and Extreme Learning Machines [11018-37]</td>
</tr>
<tr>
<td>1101814</td>
<td>A comparative study of conventional and deep learning approaches for demosaicing Mastcam images [11018-38]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Session 10</th>
<th>Signal and Image Processing, and Information Fusion Applications III</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101815</td>
<td>Perceptually lossless compression of Mastcam images with error recovery [11018-39]</td>
</tr>
<tr>
<td>1101816</td>
<td>Fusion of landsat and worldview images [11018-40]</td>
</tr>
<tr>
<td>1101817</td>
<td>Ground object detection in worldview images [11018-41]</td>
</tr>
<tr>
<td>1101818</td>
<td>Challenges in object detection in above-water imagery [11018-42]</td>
</tr>
<tr>
<td>1101819</td>
<td>Calibration and synchronization of geospatial metadata for aerial sensor systems [11018-43]</td>
</tr>
</tbody>
</table>
SESSION 11  SIGNAL AND IMAGE PROCESSING, AND INFORMATION FUSION APPLICATIONS IV

11018 1B  Extrinsic parameter calibration of 2D LiDAR-camera using edge matching and removal of infrared cut filter [11018-45]

11018 1C  Radar applications of quantum squeezing [11018-46]

11018 1D  An adaptive smooth variable structure filter based on the static multiple model strategy (Rising Researcher Paper) [11018-47]

11018 1E  Quantum radar, quantum networks, not-so-quantum hackers (Invited Paper) [11018-48]

11018 1F  A secure adaptive beamforming mechanism exploiting deafness in directional beamforming MANET [11018-49]

11018 1G  The design of a portable fusion system of an uncooled infrared bolometer and a CCD camera [11018-50]

11018 1H  Neural connectivity analysis by using 3D TMS-EEG with source localization and sliding window coherence techniques [11018-51]

POSTER SESSION

11018 1I  A novel maximum likelihood target detection with antenna selection for active phase array radar system [11018-52]

11018 1K  A multiple model adaptive SVSF-KF estimation strategy [11018-55]

11018 1M  Measuring and monitoring the QoS and QoE in software defined networking environments [11018-57]
Authors

Numbers in the index correspond to the last two digits of the seven-digit citation identifier (CID) article numbering system used in Proceedings of SPIE. The first five digits reflect the volume number. Base 36 numbering is employed for the last two digits and indicates the order of articles within the volume. Numbers start with 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B...0Z, followed by 10-1Z, 20-2Z, etc.

Ayhan, Bulent, 17
Babbitt, Sarah, 18
Balaji, Bhashyam, 18, 1C, 1E
Barr, Jordi, 07
Bar-Shalom, Yaakov, 02, 03, 04, 05, 06
Ben-Dov, R., 02
Ben-Dov, Ronen, 05
Benyamin, Minas, 0Z
Bhat, Aneesh, 0M
Bienenstock, Elisa J., 0K
Blasch, Erik, 0Q
Carlotto, Mark J., 0V, 10
Chen, Jiangnan, 1G
Chen, Yu, 0L
Choa, Fow-Sen, 1H
Chong, Chee-Yee, 0P, 0Q
Chou, Bryan, 14, 15, 16, 17
Couwenhoven, Doug W., 12
Danelakis, Antonios, 0O
Daum, Fred, 01
Daveas, Stelios, 0O
Dey, D., 04
Du, Xiaoming, 1H
Du, Zhenwei, 1G
Dzikowicz, Benjamin, 0B
Eggleton, Charles, 1K
Elmore, Paul, 0Y
Erdnüs, B., 19
Fenstermacher, Laurie, 0S
Finelli, Andrew, 02, 03
Fitwi, Alem, 0L
Fu, Rongguo, 1G
Gadsden, S. Andrew, 1D, 1K
Gatsak, Tanya, 18
Gilmour, Elizabeth, 0Y
Goldman, Robert J., 0Z
Goodman, Jacob M., 0K
Gresak, Erik, 1M
Grewe, Lynne L., 0Q, 0W, 0X
Gribben, David, 17
Grigoradis, Athanassios, 0N
Grubesic, Tony H., 0K
Gupta, Deepa, 1H
Hagen, Leif, 17
He, Miao, 1G
Himed, Braham, 03
Hiscocks, Steven, 07
Hong, Elliot, 1H
Hu, Chengzhi, 0W
Hu, Shuowen, 0Z
Huang, Jim, 0I
Huang, Kuei-Jang, 1I
Inzillo, Vincenzo, 1F
Jalowiczor, Jakub, 1M
Jimenez, Jorge, 08
Jouy, Ismail, 13
Kadar, Ivan, 0P, 0Q
Kaleda, Kelly, 0S
Kalvoda, Brian, 0T
Kim, Deokkyu, 1B
Kim, Sungho, 1B
Kirkland, David, 07
Koperski, Krzysztof, 16, 17
Kowalski, Michalen, 05
Kuppasamy, Aravinth, 0M
Kwan, Chiman, 14, 15, 16, 17
Larkin, Jude, 15
Larson, Kathleen G., 0S
Last, David, 07
Leadbetter, Eric, 0Y
Lee, Andrew, 1D
Li, Chi-Min, 1I
Li, Jiang, 16
Li, Sang Bin, 07
Luong, David, 1C, 1E
Lysehvalski, Sergey E., 0M
Mahler, Ronald, 0D
Majumder, Uttam K., 0Q
McMahon, James, 08
Milgrom, Benny, 02, 05
Nebrich, Mark, 10
Nelson, Jake R., 0K
Nock, Kristen, 0Y
Noussin, Arjang, 0I
Oxley, Mark E., 0G, 0H
Palladino, Anthony, 0K
Palmieri, Francesco A. N., 03
Paneli, V. Nicholas, 0A
Perez, Daniel, 16
Petry, Frederick, 0Y
Pouraimis, Georgios, 0N
Ptucha, Raymond, 0M
Quinta, Alfonso Ariza, 1F
Ramirez, David, 10
Rashid, Mamoon, 07
Rea, Charles, 09
Rezac, Filip, 1M
Rizogiannis, Constantinos, 0B

vii
Conference Committee

Symposium Chairs
   Jay Kumler, JENOPTIK Optical Systems, LLC (United States)
   Ruth Moser, Air Force Research Laboratory (United States)

Symposium Co-chair
   John Pellegrino, Electro-Optical Systems Laboratory, Georgia Institute of Technology (United States)

Conference Chairs
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)
   Erik P. Blasch, Air Force Research Laboratory (United States)
   Lynne L. Grewe, California State University, East Bay (United States)

Conference Co-chairs
   Bhashyam Balaji, Defence Research and Development Canada (Canada)
   Thia Kirubarajan, McMaster University (Canada)
   Ronald P. S. Mahler, Random Sets, LLC (United States)

Conference Program Committee
   Mark G. Alford, Air Force Research Laboratory (United States)
   William D. Blair, Georgia Tech Research Institute (United States)
   Mark J. Carlotto, General Dynamics Advanced Information Systems (United States)
   Alex L. Chan, U.S. Army Research Laboratory (United States)
   Kuo-Chu Chang, George Mason University (United States)
   Chee-Yee Chong, Chong Consulting (United States)
   Marvin N. Cohen, Georgia Tech Research Institute (United States)
   Frederick E. Daum, Raytheon Company (United States)
   Jean Dezert, The French Aerospace Laboratory (France)
   Mohammad Farooq, AA Scientific Consultants Inc. (Canada)
   Laurie H. Fenstermacher, Air Force Research Laboratory (United States)
   Charles W. Glover, Oak Ridge National Laboratory (United States)
   I. R. Goodman, Consultant (United States)
   Michael L. Hinman, Independant Consultant (United States)
   Jon S. Jones, Air Force Research Laboratory (United States)
   Georgiy M. Levchuk, Aptima, Inc. (United States)
Martin E. Liggins II, Indepandant Consultant (United States)
James Llinas, University at Buffalo (United States)
Uttam Majumder, Air Force Research Laboratory (United States)
Raj P. Malhotra, Air Force Research Laboratory (United States)
Alastair D. McAulay, Lehigh University (United States)
Raman K. Mehra, Scientific Systems Company, Inc. (United States)
Harley R. Myler, Lamar University (United States)
David Nicholson, BAE Systems (United Kingdom)
Les Novak, Scientific Systems Company, Inc. (United States)
John J. Salerno Jr., Harris Corporation (United States)
Robert W. Schultz, Indepandant Consultant (United States)
Andrew G. Tescher, AGT Associates (United States)
Stelios C. A. Thomopoulos, National Center for Scientific Research Demokritos (Greece)
Wiley E. Thompson, New Mexico State University (United States)
Shanchieh Jay Yang, Rochester Institute of Technology (United States)

Session Chairs

1 Multisensor Fusion, Multitarget Tracking and Resource Management I
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)
   David Luong, Defence Research and Development Canada (Canada)

2 Multisensor Fusion, Multitarget Tracking and Resource Management II
   David Luong, Defence Research and Development Canada (Canada)
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)

3 Information Fusion Methodologies and Applications I
   Ronald P. S. Mahler, Random Sets LLC (United States)

4 Information Fusion Methodologies and Applications II
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)
   Chee-Yee Chong, Chong Consulting (United States)
   David Luong, Defence Research and Development Canada (Canada)

5 Information Fusion Methodologies and Applications III
   David Luong, Defence Research and Development Canada (Canada)
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)
   Chee-Yee Chong, Chong Consulting (United States)
6 Information Fusion Methodologies and Applications IV
Chee-Yee Chong, Chong Consulting (United States)
Ivan Kadar, Interlink Systems Sciences, Inc. (United States)
David Luong, Defence Research and Development Canada (Canada)

7 Information Fusion Methodologies and Applications V
David Luong, Defence Research and Development Canada (Canada)
Ivan Kadar, Interlink Systems Sciences, Inc. (United States)
Chee-Yee Chong, Chong Consulting (United States)

8 Signal and Image Processing, and Information Fusion Applications I
Lynne L. Grewe, California State University, East Bay (United States)
Alex L. Chan, U.S. Army Research Laboratory (United States)
Mark J. Carlotto, General Dynamics Mission Systems (United States)

9 Signal and Image Processing, and Information Fusion Applications II
Mark J. Carlotto, General Dynamics Mission Systems (United States)
Lynne L. Grewe, California State University, East Bay (United States)
Alex L. Chan, U.S. Army Research Laboratory (United States)

10 Signal and Image Processing, and Information Fusion Applications III
Mark J. Carlotto, General Dynamics Mission Systems (United States)
Lynne L. Grewe, California State University, East Bay (United States)
Alex L. Chan, U.S. Army Research Laboratory (United States)

11 Signal and Image Processing, and Information Fusion Applications IV
Alex L. Chan, U.S. Army Research Laboratory (United States)
Lynne L. Grewe, California State University, East Bay (United States)
Mark J. Carlotto, General Dynamics Mission Systems (United States)
Introduction to the Invited Panel Discussion

Machine Learning in/with Information Fusion and Understanding

Information fusion provides powerful methods to integrate information and reduce uncertainty. As methods of big data, artificial intelligence/machine learning (AI/ML), and network systems increase in complexity, information fusion provides opportunities for scalable solutions. Information fusion methods support dynamic processing, data coordination, and complexity management. Understanding; including recognition, localization and interaction; is brought to new levels of achievement with the combination of machine learning and sensor/information fusion, which provides situation understanding.

For example, areas of interest include: (1) current state of ML, which can support information fusion while it can provide understanding; (2) applications yielding urban automation such as: transportation, mobility and smart cities; (3) ML effects on communications and infrastructure examples such as: physical networks, social networks and cyber-communication networks; and (4) issues of security, privacy, trust, scalability, regulation and dependency.

The objective of this panel is to foster methods and discussion on how information fusion, understanding, and machine learning can support real world needs though potential approaches and addressing challenges. A number of invited experts discussed challenges in processing and research, and addressed these challenges with Machine Learning and IF. The panel with highlighted impending issues and challenges and illustrated both conceptual and real-world related examples associated with real applications.

Chee-Yee Chong  
Lynne Grewe  
Ivan Kadar  
Erik Blasch
Invited Panel Discussion Machine Learning in/with Information Fusion and Understanding

Organizers
Chee-Yee Chong, Independent Consultant
Lynne Grewe, California State Univ. (U.S.A)
Ivan Kadar, Interlink Systems Sciences, Inc.
Erik Blasch, Air Force Research Lab

Moderators
Ivan Kadar, Interlink Systems Sciences, Inc.
Chee-Yee Chong, Independent Consultant

April 15, 2019
SPIE Conference 10118
“Signal Processing, Sensor Fusion and Target Recognition XXVIII”
Baltimore, MD, 15-17 April 2019

Invited Panel Discussion

Panel Participants:
Dr. Erik Blasch, Air Force Research Lab., U.S. A.
Dr. Chee-Yee Chong, Independent Consultant, U.S.A.
Professor Lynne Grewe, California State Univ, East Bay, U.S.A.
Professor Paul Bendich, Duke Univ., Durham, NC, U.S.A.
Dr. Ivan Kadar, Interlink Systems Sciences, Inc., U.S.A.
Professor Vladimir Pavlovic, Rutgers Univ., New Jersey, U.S.A.
Professor Edward L. Waltz, Naval Postgraduate School Center for MultiINT Studies, Monterey, CA, U.S.A.
Invited Panel Discussion

Presentation Topics

“Machine Learning (ML) to Support Information Fusion”
Professor Edward L. Waltz, Naval Postgraduate School Center for MultiIN T Studies

“InTEGRATING Deep Learning with Models for Information Fusion”
Dr. Chee-Yee Chong, Independent Consultant

“Application of Machine Learning/Information Fusion to Infrastructures”
Dr. Erik Blasch, Air Force Research Lab.

“Information Fusion & ML for Social Signal Processing”
Professor Vladimir Pavlovic, Rutgers Univ.

Professor Lynne Grewe, California State Univ, East Bay, CA

“Topology and Geometry for Tracking and Sensor Fusion”
Professor Paul Bendich, Duke Univ. Durham NC

“Perspectives: Why use Maximum Entropy (MaxEnt) Machine Learning in Information Fusion”
Dr. Ivan Kadar, Interlink Systems Sciences, Inc
Machine Learning (ML)
Support to Information Fusion

Ed Waltz
Naval Postgraduate School
Center for MultiINT Studies

SPIE Defense + Commercial Sensing 2019  April 15, 2019

Topics

- Key Terms
- Data Mining and Data Fusion (1998)
- Machine Learning and Information Fusion (2019)
- Applying the Learned Patterns in Information Fusion
- Three Categories of ML integration with Information Fusion
- The Next Steps
**Key Terms**

- **Data Mining** – the process of analyzing data in large data sets by means of correlation and automated induction to discover useful patterns.
- **Machine Learning** – A set of AI process that performs automated induction of models of patterns in data to enable detection (prediction), classification and clustering of similar data.
- **Automated Induction** – Statistical processes that induce a generalized model that represents a particular set of data.
- **Process Adaptation** – the ability of a system to monitor its operations, distinguish changes in the operating environment, and adapt processes to continue to perform the design functions in the presence of changes.
- **Cyberspace** is a global domain within the information environment consisting of the interdependent networks of information technology infrastructures and resident data, including the Internet, telecommunications networks, computer systems, and embedded processors and controllers (JP 3-12[R]).

---

**Data Mining and Data Fusion (1998)**

- Manual use of automated induction tools to induce patterns of similarity (clusters)
- Integration with data fusion was manual
Machine Learning and Information Fusion (2019)

- **Model learning phase**
  - Object Models
  - Graph Models
  - Sequence Models
- **Streaming and Batch “Prediction”**
- Figure Source: [2] Ed Waltz, *Intelligence Integration and Automation*, National Intelligence University Press, forthcoming 2019

Applying the Learned Patterns

- Three categories of learned subjects:
  - Discrete Objects
  - Relationships between objects
  - Behavior of Objects
- Derived causal explanations and narrative descriptions

Figure Source: [2] Ed Waltz, *Intelligence Integration and Automation*, National Intelligence University Press, forthcoming 2019
### Categories of ML-IF (Information Fusion) Integration

<table>
<thead>
<tr>
<th>Category</th>
<th>Machine Learning Role</th>
<th>Example Applications [Sources]</th>
</tr>
</thead>
</table>
| 1 ML Pattern Learning for Detection - Classification | • LEARN a signature model of a target subject by supervised learning process  
• APPLY the model to detect and classify future occurrences | • Learn fracking pad pattern for subsequent automated trend monitoring [3]  
• Learn Ship signatures for maritime tracking [4] |
| 2 ML Pattern Learning For Discovery | • LEARN normalcy model of a subject by supervised learning or unsupervised clustering  
• APPLY the model to discover anomalies and incongruities | • Detect and Discover new patterns of social disruption by comparing social activity to ensembles of ML models[3]  
• Discover anomalous behavior for insider threat behavior [5] |
| 3 ML Process learning to recommend Process Adaptation | • LEARN model of DF processes (Collection, Detection, Discovery) and classify performance  
• APPLY model to measure values and recommend adaptations to processes; | • Learn best performance cases in a large-scale process and recommend new candidates and parameters [6] |

### Urban Automation Applications

- Transportation, mobility and smart cities
  - **Traffic** Learning and flow prediction (emergent traffic jams; high-risk accident situations, weather-induced traffic patterns); abnormal behavior detection
  - **Energy** load learning and predictive adaptation
  - **Crime** traffic pattern learning (Crime mapping and analysis)

- Communications and infrastructure
  - **Social** Media learning and social disruption detection (Combined with traffic)
  - **Cyber** behavior (commerce, financial, criminal) learning and detection of emergent patterns of use of cyber communication channels

Always consider multiple source machine learning (e.g. across traffic, social, cyber)
Key Steps to ML-IF Integration

• Move from Labelling Data to Labelling Results
  – Implement Cat 3 ML capabilities by measuring the relative value of information fusion results
  – Example: Apply ML to explicitly measure all products, estimate value

• Develop the New Phenomenology
  – Develop the causal theories of combined physical, cyber, social phenomena
  – Example: What is the social, cyber, and traffic behavior of a bank robbery team prior to and after the robbery event? How is it learned?

• Beware the issues of Learning Bias and Concept Drift
  – Develop procedures to measure bias and drift, consider the vulnerabilities to self-deception and adversary manipulation
  – Example: Consider how learning bias, introduced by limited sampling, may introduce prediction errors in robbery surveillance?

References

ABSTRACT

Machine learning (ML) technology develops computational models that are used in the information fusion (IF) process to perform detection, classification and prediction. ML requires large, labelled training data sets to develop boundary conditions to learn the mathematical manifolds to accurately characterize labelled data with minimum bias. The integration of ML (to learn models of objects, relationships and behaviors) and IF (to apply the models for estimation, detection and prediction) enables the automated understanding of complex human activities in urban areas. This brief paper describes the integration of data fusion and machine learning technologies and explains applications in urban areas where large volume data exists and ML-IF integration can provide anticipation and warning of urban events. Approved for public release: distribution unlimited.

Keywords: Machine Learning, Information Fusion, Automated Induction, Urban Understanding

1. INTRODUCTION

By the mid-1990’s the manual use of automated induction tools (called data mining) to induce patterns of similarity (clusters) and outliers was integrated with data fusion for applications requiring rapid discovery of new patterns (threats) and immediate use of the pattern model in the fusion process. [1] The integration was loose; pattern discovery was offline and threat models learned by analysts were introduced manually. The past decade’s success in apply much deeper learning methods (e.g. convolutional neural nets with large scale training sets that produce very complex manifolds to describe training clusters) has enable more automated integration of ML into the IF process.

2. LEARNED MODELS IN INFORMATION FUSION

The integration of ML to provide models can occur at any the three IF upstream levels: Detection and classification of objects at level 1 (physical, cyber or other discrete things or events), related sets of objects (e.g. the learning of graph models), or the behavior of either over time. The integration (Figure 1) occurs when machine learned models are applied the level 1, 2 or 3 processes of IF. Current ML processes require monitoring to detect concept drift (where change in phenomena in the real world that may require new training and model update due to changes in statistical properties) or bias (where model results are systematically prejudiced due to errors of assumption or training data). The feedback loop from IF must monitor for model performance at all levels. IF benefits the ML process by the ability to make sense of phenomena across multiple sources, detect anomalies, and provide greater assurance that drift and bias are not present.

Figure 1. Exploratory Data Analysis (EDA) and Machine Learning (ML) provide the models for the subsequent Information Fusion (IF) process to perform detection (prediction) classification or estimation. [2]
Consider three categories of ML applications to integrate with IF (Table 1). The first, most common, is the learning of signatures (e.g. social disruption patterns) for IF to anticipate or detect. The next is learning of “normalcy” patterns to distinguish anomalies and outliers that may represent new behaviors, threats, or opportunities. The third application category is learning of the IF process itself, to refine and adapt the IF process. For this, quantitative measures of IF performance are required to properly label training data sets.

Table 1. Three categories of Machine Learning (ML) roles with example applications

<table>
<thead>
<tr>
<th>Category</th>
<th>Machine Learning Role</th>
<th>Example Applications [Sources]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ML Pattern Learning for Detection - Classification</td>
<td>• LEARN a signature model of a target subject by supervised learning process • APPLY the model to detect and classify future occurrences</td>
<td>• Learn fracking pad pattern for subsequent automated trend monitoring [3] • Learn commercial ship signatures for maritime ship tracking [4]</td>
</tr>
<tr>
<td>2 ML Pattern Learning For Discovery</td>
<td>• LEARN normalcy model of a subject by supervised learning or unsupervised clustering • APPLY the model to discover anomalies and incongruities</td>
<td>• Detect and Discover new patterns of social disruption by comparing social activity to ensembles of ML models [3] • Discover anomalous behavior for insider threat behavior [5]</td>
</tr>
<tr>
<td>3 ML Process learning to recommend Process Adaptation</td>
<td>• LEARN model of Information Fusion (IF) processes (Collection, Detection, Discovery) and classify performance • APPLY model to measure values and recommend adaptations to processes</td>
<td>• Learn best performance cases in a large-scale process and recommend new candidates and parameters [6]</td>
</tr>
</tbody>
</table>

3. URBAN APPLICATIONS

Smart cities have ubiquitous sensors and networked communications that enable learning of the behavior patterns of dense human populations, and the use in information fusion systems to improve safety, efficiency and regulation. Consider example applications in two areas. In the first area, transportation and energy will benefit from traffic pattern learning and flow prediction to recognize emergent traffic jams, high-risk accident situations, abnormal traffic, and emergent weather-induced traffic patterns). Energy load learning will aid predictive adaptation for power distribution, and crime traffic pattern learning is already being used by law enforcement in crime mapping and analysis. The second area that will benefit is communications and infrastructure monitoring in which ML models are already providing warning of social disruptions (riots, protests, mass gatherings, flash mobs, etc.) by monitoring patterns in social media. Social media data combined with traffic data can provide further understanding of these urban disruptions and the best approaches to disperse them. ML models of cyber behavior can benefit an understanding and warning of commerce, financial, and criminal activities that rely on the urban communication infrastructure.

REFERENCES

Motivation

- Humans are good at sensor and data fusion
  - Low level perception from sight and sound
  - High level reasoning to understand situation, cause and effect
- Artificial intelligence (AI) attempts to replicate human sensor and data fusion capabilities
- Model-based (AI and non-AI) approaches rely on domain models
  - Represent and reason with models
  - Models provide explainable results
- Deep learning (deep neural networks) relies only on data
  - Neural networks learned from large amounts of data
  - Easy implementation and excellent performance in some applications but results are hard to explain
- Robust information fusion and understanding should use both deep learning and model-based reasoning
Moving and Stationary Target Acquisition and Recognition (MSTAR) – Model-Based Approach

- Program manager said "neural network is not allowed"

MSTAR Target Types and SAR Images

U. K. Majumder, "Deep Learning in AI and Information Fusion" SPIE Panel Discussion, 2018, Orlando, FL
Deep Learning for MSTAR Using Caffe

- Deep learning MSTAR is much easier to develop than model-based MSTAR
- Performance is also very good (99% accuracy) for test data
- However, it is difficult to modify the network for other operating conditions with camouflage and deception

Face Grouping by Google Photos - Adults

Adult faces are correctly grouped into a “track” over time
• Faces of two babies are incorrectly grouped
• Incorrect grouping can be detected easily from
  • Metadata (dates when photos are taken)
  • Knowledge/model that babies grow fast and appearance change quickly

Model-Based versus Learning (Function)-Based AI

- Explicit problem knowledge
- Represent and reason
- Transparent understandable processing

- Problem knowledge captured by data
- Train neural network from data
- Transparent processing but hard to understand or explain
Why Explainable Algorithms are Important

• Assumptions provide operating conditions for algorithm
• Reasoning/processing approach allows generalization to other operating conditions
• Explainable algorithm model supports prediction of performance, e.g., error bounds from Kalman filter equations
• Fusion by other nodes requires models, e.g., 2D video tracks with other 2D video tracks
• Critical decisions have to be explained, e.g., right to explanation in EU General Data Protection Regulation (GDPR)
• Good engineering requires design review, not possible without explainable components

Robust Information Fusion

• “Robust” is hard to define precisely, but is opposite to “brittle”
• A trusted system has to be robust
• Properties of robust information fusion
  • Well-defined operating conditions
  • Predictable performance
  • Insensitivity to small changes
  • Graceful performance degradation within operating condition
  • Adaptability to context
• Robust information fusion requires components that are either
  • Explainable with models
  • Tested and statistically characterized with large amounts of data
Proposed Approach for Integrating Deep Learning and Model-Based Reasoning

- Map processing approach to required ability
  - Deep learning for low-level functions such as perception (animal-like abilities*)
  - Model-based reasoning for high level situation assessment (human-level intelligence*)
- Design rule
  - Model rich, data rich – model-based reasoning
  - Model rich, data poor – model-based reasoning
  - Model poor, data rich – deep learning
  - Model poor, data poor – get better sensors

*A. Darwiche, “Human-Level Intelligence or Animal-Like Abilities?” CACM, Oct. 2018

Allocating JDL/DFG* Fusion Levels to Learning and Models

- Level 0: DL is suitable for speech recognition, feature extraction
- Level 1: DL can recognize objects/person and skip level 0; tracking can use both DL and model-based reasoning
- Level 2 and above: model-based reasoning (users want explanations)

*JDL/DFG – Joint Director of Laboratories/Data Fusion Group
Multiple RNNs for Online Multiple Object Tracking

- RNN for prediction and update, target birth and death
  - Completely model free
  - Handles arbitrary dynamics, distributions
- LSTM for data association
  - Handles combinatorial problem
  - Controls memory in RNN


RNN and LSTM for Multiple Object Tracking

- Model-free approach is hard to explain and adapt to different situations
Feature-Aided Tracking using Deep Learning and Models

• Possible associations are found by deep learning
• Association costs are computed using models (object dynamics)
• Target tracks are computed by solving optimization problem on association graph


Conclusions

• Deep learning performs well in some essential fusion functions (e.g., speech recognition, image recognition) but not suitable for other functions
• Model-based reasoning is needed for high-level fusion but has limited success with unstructured high-dimensional data, e.g., image, speech
• Robust fusion systems should use
  • Models when they are available and can be represented for reasoning
  • Deep learning when models are weak and data are available
• Successful use of deep learning in information fusion requires
  • Understanding capabilities and limits of function-based approaches
  • Characterizing deep learning algorithms (assumptions, operating conditions, and performance) for use by other fusion functions
Application of Machine Learning/Information Fusion to Infrastructures

Erik Blasch: AFOSR

Outline

• Context Assessment
  – Use of domain information
  – Multi-domain awareness

• Dynamic Data Driven Applications Systems (DDDAS)
  – Data Augmented with Models
  – Multi-domain (Space, Air, Cyber)

• AI/IF “in Use”
  – Combine context/models
  – Deliver to the user
Dynamic Data Analysis

- Dynamic Data Driven Applications
- DDDAS for MDC2 Awareness
- DDDAS Projects
  - A/D in Motion
  - A/D at Rest
  - A/D in Use
- Examples

Dynamic Data Analysis Areas:

- Application Algorithms
  - Statistical Algorithms (e.g., Data Science)
- Instrumentation Methods
  - Measurement Systems (e.g., Data Collection)
- Modeling and Simulation
  - High dimensional Modeling (e.g., Big in Use)
- Systems Software
  - Real-time Development (e.g., Data Computing)
DDDAS Interactive Functions

DDDAS Methodology

- Sensing Measurements
- Methods
- Modeling

- Run Time Architecture
  - Statistical Learning
  - Instrumentation Control
  - User Refinement

- Applications Modeling
  - Algorithms Selection
  - Operating Conditions

- Situation Simulation

DDDAS Architecture

DDDAS augments data through simulation for estimation
# Handbook of DDDAS

## Literature Review

<table>
<thead>
<tr>
<th>Awareness</th>
<th>Monitoring</th>
<th>Data Fusion</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scientific Theory</strong></td>
<td>Weather Forecasting</td>
<td>Wildfire Monitoring</td>
<td>Ash Detection</td>
</tr>
<tr>
<td><strong>Domain Method</strong></td>
<td>Space Awareness</td>
<td>Structural Health</td>
<td>State Estimation</td>
</tr>
<tr>
<td><strong>Architecture Design</strong></td>
<td>Network Trust</td>
<td>Energy Analysis</td>
<td>Image Computing</td>
</tr>
</tbody>
</table>


## Challenges

![Challenges Diagram](https://via.placeholder.com/150)

**Instrumentation Methods**

**Data Analytics**

**Real World Applications**

**Theory**

**Fidelity Challenge**

**Matching Challenge**

**Content Modeling**

**Sampling Challenge**

**Computation**

**Systems Software**

**Algorithm**
OBJECTIVE:
Identify AF relevant projects that span the research objectives though novelty, innovation, and verification/validation to provide context awareness

RESEARCH AREAS:

- **Application Algorithms**
  - Statistical Algorithms (e.g., Data Science)
- **Instrumentation Methods**
  - Measurement Systems (e.g., Data Collection)
- **Modeling and Simulation**
  - High-dimensional Modeling (e.g., Data in Use)
- **Systems Software**
  - Real-time Development (e.g., Data Computing)

**DDDAS Overview**

Innovations in Dynamic Data Science
DDDAS uses simulated data from a high-dimensional models to augment measurement systems leveraging statistical methods, simulation, and computation architectures.

www.1dddas.org

Swarms: Distributed ML

Multi-Domain C2
Control Diffusion

Man-Machine Analytics
Operator Infusion

ISR
Information Fusion

Action-to-effect Models
Decison-Support Models
Contextual Models

Autonomy in MOTION
Autonomy in USE
Autonomy at REST

Agent Based Distributed Fusion

Multi-Domain Command and Control
Science of measurements, sensing, and estimation

Program Objectives:
To innovate, manage, and design physics-based theoretical methods for sensing and estimation for environment, context, situation, and energy-aware systems which require advances in high-dimensional modeling, nonlinear estimation, data assimilation, uncertainty quantification, multi-scale data fusion, and data management architectures

Research Areas:
- Data Assimilation and Sensing
  - High-fidelity Models
  - Reduced-Order Modeling
  - Nonlinear Estimation
- Context Awareness and Sensing
  - Sensor Processing & Mgt
  - Self-healing Systems
  - Autonomous Robotics
- Data Flow design, optimization, & security
  - Cybersecurity
  - Energy/Power Grids
  - Agile, Multi-domain Sys

Environment Awareness
Situation Awareness
Cyber Awareness
Information Fusion for Infrastructure Analysis

Erik Blasch
Air Force Research Laboratory, Arlington, VA, 22203

ABSTRACT

Complex systems are composed from various networks such as physical, computer, cyber, communication, cognitive, and social systems. Information fusion can support complex systems uncertainty analysis for various infrastructures applications such as electrical power grids, human-machine coordination, and urban systems. In addition, advances in artificial intelligence (AI), machine learning (ML), and data mining; along with information fusion provide a big data approach for multidomain complex systems analysis. The presentation position focuses on the Dynamic Data Driven Applications Systems (DDDAS) paradigm that brings together high-dimensional network models, real-time data measurement processing, and uncertainty quantification to support infrastructure network analysis. The challenges of applying information fusion to infrastructure system include multi-dimensional models, asynchronous measurement updates, human-machine architectures, communication protocols, and user awareness support.

Keywords: Information fusion, big data, complex systems, Dynamic Data Driven Applications Systems

1. INFRASTRUCTURE ANALYSIS

Information fusion seeks to reduce uncertainty [1] and previous discussions demonstrate the coordination between big data, artificial intelligence/machine learning (AI/ML), and network systems [2]. Complexity management, especially for prediction control of an infrastructure (e.g., facilities and enterprises) requires models, measurements, and estimation techniques [3]. Different types of networks include: (1) physical networks: buildings, roads, and sensors; (2) social networks: organization relations, semantic discussion, and business transaction; and (3) cyber-communication networks: power grids, data transfer, and energy distribution. Recent advances in Dynamic Data Driven Applications Systems (DDDAS) have demonstrated methods towards targeting [4] and infrastructure analysis [5] as shown in Table 1. The DDDAS paradigm brings together scientific theory through first-principles modeling, domain adaptation, and architecture design. Since a complex system is difficult to monitor, DDDAS uses simulations for augmented measurements (Fig. 1) as well as run-time architectures for analysis (Fig. 2) [6].

Table 1: Infrastructure Analysis using the Dynamic Data Driven Applications Systems (DDDAS) Paradigm.

<table>
<thead>
<tr>
<th>Scientific Theory</th>
<th>Monitoring</th>
<th>Data Fusion</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weather Forecasting</td>
<td>Wildfire Monitoring</td>
<td>Ash Detection</td>
<td>Medical Support</td>
</tr>
<tr>
<td>Space Awareness</td>
<td>Structural Health</td>
<td>State Estimation</td>
<td>Self-Aware Vehicles</td>
</tr>
<tr>
<td>Network Trust</td>
<td>Energy Analysis</td>
<td>Image Computing</td>
<td>Cyber-Physical Systems</td>
</tr>
</tbody>
</table>

Figure 1: DDDAS Model.     Figure 2: DDDAS processing.
Networks comprise a variety of interlinked systems that require effective, efficient, trusted, and scalable data analytics from which information fusion can support systems control and information exploitation [7]. Coupling low-level information fusion (e.g., target tracking [8], text analytics [9], and image fusion [10]) with high-level information fusion (e.g., situation assessment [11], threat assessment [12], and user culture [13]) requires advances in modeling [14], visualization [15], and context processing [16]. DDDAS brings together advances in theory, measurements, and software; enabling the dynamic interaction between data, models, control, and analytics.

A key issue is the types, fidelity, and relevance of multi-dimensional models to support the infrastructure analysis. Figure 3 presents a multidomain example of an infrastructure that links space, air, and cyber systems. The network infrastructure can leverage advances in information fusion, AI, and ML for coordinated control. Each domain can utilize knowledge from another domain (e.g., consumer) and cue another domain (e.g., producer). The coordination between the domains would vary among the graphical, nonlinear, and estimation fusion techniques selected.

Another challenge would be big data processing of the entire system for efficient, effective, and robust operations. One issue to acknowledge is that each domain and subsequent measurements are updated asynchronously. The measurement updates are at different rates which might require different information fusion techniques to support the operations in each domain. Each domain would have specialized users and dedicated machines that need to be coordinated.

A systems infrastructure comprises different machines to include cloud, fog, and edge nodes. To control a systems-level infrastructure comprises a distributed nature of multiple agents operating over different visualizations and computational tools towards a comprehensive understanding of the evolving scenario. An example of complex systems comprises information fusion (IF), operator infusion (OI), and control diffusion (CD) agents as shown in Figure 4. Inherently, there are different interactions of the agents with four options: (1) operator node in which a distributed user combines IF/OI/CD results, (2) user/cloud node that delegates control to different information fusion tasks, (3) fog node such as pilot that coordinates information fusion and diffusion control, and finally an (4) edge node in which devices feed data to a person that fuses the data.

To coordinate the infrastructure would be of different communication protocols. An example of a future infrastructure is a distributed set of internet of things (IoT) as edge nodes with that of a cyber-physical system (CPS) centralized processing. Since the system is multidomain, then the design of each domain has a different development protocols that have standards, metrics, and mandates to follow when linking the systems of different domains. The interaction between domains would foster a discussion and analysis on a joint communication system that seek a benefit (and balance) for each system within the infrastructure.

Finally, as the infrastructure is essentially a large machine that enhances a society function, there are various users empowered to monitor and control the system. Hence, the future methods for infrastructure situation awareness extend from a single domain to that of a joint understanding of the other domains. Clearly, a user would not be able to track...
each IoT device, nor be able to understand the big data processing of a cloud server, so methods of sampling, visualization, and systems performance need to be abstracted to a level commensurate with the cognitive processing of humans. Likewise, the coordination between human users has to be in a form for coordinated control and analysis. The coordination is especially needed when there is a problem or emergency from which the users need to take action (e.g., power grid failure or attack). For a given network attack [17], the machines and users need to manage the scenario based on the operating conditions [18]. Recent examples include methods for IoT control, security, and access [19, 20], that highlight the complexities in supporting robust infrastructure operations.

2. CHALLENGES FOR INFRASTRUCTURE ANALYSIS

The challenge for infrastructure analysis is that it requires high-dimensional analysis where modeling of the systems of systems is rather unknown and difficult to assess. An example is the inclusion of the Internet of Things (IoT) as edge devices that coordinate with large infrastructure situations as per cyber-physical systems. The system requires different agents to support multidomain coordination. The challenges include:

- **Multi-dimensional models** requiring methods for multi-resolution estimation updates at different rates;
- **Asynchronous processing** of distributed measurements through the infrastructure;
- **Machine architectures** that leverage high performance computing for cloud, fog, and edge devices.
- **System of Systems communication** when combining multi-domain infrastructure entities;
- **User awareness** methods allowing for performance visualization, assessment, and understanding.

In summary, information fusion can support infrastructure control and assessment challenges from which data-driven machines and knowledge-driven user support approaches must be coordinated to meet complex systems analysis.

**Acknowledgements**

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Government.

**REFERENCES**

Information Fusion & ML for Social Signal Processing
Professor Professor Vladimir Pavlovic, Rutgers Univ.

Please note:

Presentation is available to download:
https://www.dropbox.com/s/22aqr0ir6fifdbq/spie_apr19.pdf?dl=0

Movies associated with the presentations are available in the links below
https://www.dropbox.com/s/4511ui4glwr9qfy/spie_apr19_small.m4v?dl=0
Powerpoint show version, about equally large, with videos as well:
https://www.dropbox.com/s/tdqvd111fkadaif/spie_apr19.ppsx?dl=0
Multimodal Deep Learning: Issues and Solutions for Creating Well Performing Systems

Lynne Grewe

aComputer Science, California State University East Bay, 25800 Carlos Bee Boulevard, Hayward, CA USA, 94542

Outline

1. Comparing Multimodal DL and conventional Multimodal ML
2. Multimodal Correlation / Alignment
3. The Fusion Stage & Multimodal Impacts– early, mid, late
4. Dealing with Multi-modal Data Issues (lack, missing)- Transfer Learning, Pretraining w/AutoEncoder, Perturbation, Generative Networks
Seeing Eye Drone

Mobility for low vision

ULearn
Assisted Learning

Facial Emotion Detector
Tips Generation

Camera Initialize, Continuous Capture Loop
Face Detection
Emotion Detection
Frustration Detected

Web View Interface
Parse web content and get meta-data
NLP analysis
Web Search for tips

Change URL or click on tip

Launch ULearn
Comparing Multimodal DL and conventional Multimodal ML

<table>
<thead>
<tr>
<th>Multimodal Deep Learning</th>
<th>Multimodal Conventional Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Both modality-wise representations (features) and shared (fused) representations are learned from data.</td>
<td>Features are manually designed and require prior knowledge about the underlying problem and data.</td>
</tr>
<tr>
<td>Requires little or no preprocessing of input data (end-to-end training).</td>
<td>Some techniques, like early fusion, may be sensitive to data preprocessing.</td>
</tr>
<tr>
<td>Supports early, late, or mid fusion.</td>
<td>Typically performs early or late fusion.</td>
</tr>
<tr>
<td>Fusion architecture can be learned during training</td>
<td>Rigid fusion architecture usually handcrafted.</td>
</tr>
<tr>
<td>Deeper, complex networks typically require large amounts of training data</td>
<td>May not require as much training data</td>
</tr>
<tr>
<td>Numerous hyperparameter tunings vital for state-of-art performance.</td>
<td>May not have as many hyperparameters as deep-learning architectures.</td>
</tr>
<tr>
<td>Compute intensive, requires powerful graphics processing units (GPUs) for reasonable training time.</td>
<td>GPUs may offer speed-up, but not critical.</td>
</tr>
<tr>
<td>May have faster run times</td>
<td>Run times may be slower</td>
</tr>
</tbody>
</table>
MULTIMODAL DATA ALIGNMENT/CORRELATION

Modality Data Alignment/Correlation

- Time Correlation


- Spatial Correlation

Modality Data Alignment/Correlation

- Learning Correlation ➔ applying DL to the task

Using a CNN followed by an LSTM to learn correlations for Text and Image (note can be used for text generation/captioning) where a proposed region is passed into the CNN.

---


---

Modality Data Alignment/Correlation

- Learning Correlation ➔ applying DL to the task

**Extension** – here different keywords in sentence attributed to different regions in the image

---

FUSION LEVEL AND MULTIMODAL

Where does fusion take place?

EARLY Fusion  MID Fusion  LATE Fusion
Different Systems – choosing different fusion levels

<table>
<thead>
<tr>
<th>Reference</th>
<th>Year</th>
<th>Modalities</th>
<th>Problem</th>
<th>Fusion Method</th>
<th>Model</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ngiam et al.</td>
<td>2011</td>
<td>Audio, video</td>
<td>Speech classification</td>
<td>Mid</td>
<td>Generative</td>
<td>Sparse RBM</td>
</tr>
</tbody>
</table>
| Srivastava and Solakhu
  dinov                  | 2012 | Image, text         | Image annotation                 | Mid           | Generative  | DBN          |
| Cao et al.             | 2014 | Medical images, text descriptions | Content-based medical image retrieval | Mid           | Generative  | DBM          |
| Liang et al.           | 2015 | Gene expression, DNA methylation, and drug response | Cancer subtype clustering | Mid           | Generative  | DBM          |
| Volada et al.          | 2016 | Multispectral imagery | Semantic segmentation            | Early         | Discriminative | FCNN        |
| Simonyan and Zisserm
  an                     | 2014 | Image and optical flow | Action recognition              | Late          | Discriminative | CNN          |
| Kohou et al.           | 2015 | Video, audio        | Emotion recognition              | Late          | Discriminative | CNN, RNN, SVM, and AE |
| Liu et al.             | 2015 | MRI, PET            | Medical diagnosis                | Mid           | Discriminative | Stacked AE, SVM |
| Poria et al.           | 2015 | Video, audio, text  | Sentiment analysis               | Mid & Late    | Discriminative | CNN, SVM     |
| Lenz et al.            | 2015 | Intensity, depth video | Robotic grasping                 | Mid           | Discriminative | Stacked AE and MLP |
| Jain et al.            | 2016 | Video features, GPS coordinates, vehicle dynamics | Driver activity anticipation | Late          | Discriminative | LSTM         |

Early fusion example

- 3D and 2D image, here DL Network is a CNN

![3DCNN architecture](image)

Another Early Fusion example


4D = r, g, b, depth

Another Early Fusion Example

- Here is a NON visual (but, temporal data) sensor example

GUIDELINE 1: Nature of Data Can Effect Network choices and Fusion Level – **Mid-level fusion**

- CNNs are type of DL network used for spatial data (images, etc) and not used for textual data. So, if you want a CNN and have both kids of data, early fusion is not possible.

Mid-level fusion, where first similar image based (3D and 2D) multimodal data is processed with a CNN and in parallel the skeleton data is processed with a Gaussian-Bernoulli Restricted Boltzmann Machine (DBN-Deep Belief Network) before being fused and the output used in a Hidden Markoff Model for Gesture Recognition (a temporal event).

**ANOTHER Mid-level example: separate CNN run in parallel for each sensor data**

Must combine results of separate CNNs:
- Fully connected layer
- SVM
- Rule-based / algorithmic combination


Late(r) Fusion example

• Application = learn correlations between EEG response and Image stimuli
• Multimodal data = 2D image processed with CNN and EGG (brain electroencephalogram) processed with LSTM


Another Late fusion example.


Merge later = Rulebase, SVM, Another FC Network, Algorithmic
Another Late fusion example


- 2 CNNs: 1) for single RGB image  2) for optical flow between current and previous frame(s).

GUIDELINE 2: Strong Dependency on Modality X  
HIERARCHY of Network fusion  
(modality X DL Network +  Multiple Modality DL Network)

- From – Speech Recognition: best to combine Audio DL network with Mid-level fused Multiple Modality DL Network  
WHICH option is best?


- Compared early, mid and single sensor for Person Detection & Tracking with RGB + Depth

**early:** 1 CNN for all data

**mid:** 2 CNN in parallel

**single sensor**

No clear answer -- test

Case for Mid/late fusion - data does not “align” spatially

- Example speech (not spatial, temporal only) and vision
- Process in parallel with separate CNNs
  → this works because there is temporal alignment

MULTIMODAL DATA → MORE COMPLEXITY

Problem – Deeper Network generally slower

- Resnet-152 is 5 times slower than Alexnet
GUIDELINE 1: Reduce overall DL network depth via mid- or late-fusion

- Early fusion requires processing of all multimodal data at all layers → most increased complexity over single source data DL networks.

- Late fusion has only slightly increased depth of processing each modality in its own separate single modality DL network.

- Mid fusion is partway between early and late in required depth.
Training data

- Not enough data
- Very few multimodal datasets (only audio/video, text/video and some 3D/2D).

SOLUTION -> Retraining (transfer learning)

- Are there existing trained publicly available multimodal models?

Multimodal training sets emerging

- KITTI Stereo gray- and color video, 3-D-LIDAR, inertial and GPS navigation data Autonomous driving, 2017
- FCVID Video and audio Action recognition, 2017
- MM-IMDb Video, images, and text metadata Movie genre prediction, 2017
- Oxford RobotCar Six cameras, LIDAR, GPS, and inertial navigation data Autonomous driving, 2016
- UTD-MHAD Depth and inertial sensor data Human action recognition, 2015
- ChaLearn looking at people RGB-D, audio, skeletal pose Human activity recognition, 2014
- Berkeley MHAD Multiviewpoint RGB-D and skeletal pose data Human activity recognition, 2013
- MHRI data set Chest, top RGB-D, face, video, and audio Human–robot interaction, 2016
- Multimodal BRATS T2-, FLAIR-, post-Gadolinium T1-MRI, perfusion, and diffusion MRI and MRSI Brain tumor segmentation, 2015
- H-MOG Nine smartphone sensors and interaction data Continuous authentication in smartphones, 2016
- Pinterest Multimodal Images and text (40M) Multimodal word embeddings, 2016
- MHEALTH Accelerometer, electrocardiogram, magnetometer, and gyroscopes Health monitoring, 2015
- KinectFaceDB RGB-D and facial landmarks Face recognition, 2014
- RECOLA Audio, visual, and physiological Emotion recognition, 2013
GUIDELINE 3: Multimodal – Retraining*, will generally mean mid- late- fusion

- Premise: you have an existing trained model for one of your modalities*

Create new “larger” network that reuses this model.

GUIDELINE 4: Lack of Training Data – Pre-training via AutoEncoder

- Idea here is you use auto-encoder of a unsupervised trained auto-encoder/decoder network to be the pre-trained start of your DL network

STEP 1: train using Unlabeled input data an Autoencoder/decoder

STEP 2: use trained Autoencoder as first stage of your DL network and train with your limited labeled data
PROBLEM: Lack of One (+) Modality Data (or missing)

• Scenario: If it is expensive to generate one of the modality data you need (consider something like EEG patient data)

• **Solution 1:** generate new data through perturbation process
• **Solution 2:** then you might consider creating a generative network for that modality

---

Guideline 5: solution to lack of one(+) Modality data → data perturbation

• Image resizing, moving of regions, rotations, photometric changes

• How do we do this for all different kinds of data → expert domain knowledge is minimally necessary

---

Danger?: There are some perceived dangers as you are essentially introducing potentially artificial changes
IDEA/GUIDELINE 6: solution to lack of one(+) Modality data → Cross Modal Generative Network

- If one modality is more limited, but, you still have some, you may be able to create a cross-modal generative network.

**Question:** How is the new resulting data here different than perturbation solution?

**NOTE:** Generative networks used in DL GAN (generative adversarial networks)

**Example generative data using Multimodal training**

- Create CNN network based generation of EEG (output) from input Image.

Example 2 – different interpretation of cross-modal generation

- Here you present the system with data from all modalities (here text and image) and the output is a newly generated (similar) data set.


Dangerous? Generative methods

- What do you think?

Col 1-3 RGB 2D image
Col 4-6 corresponding Depth image

Missing Modality data

Generated Modality data
Option 1 – resize data

- **Problem of variable sized input** propagates down to the first fully connected/inner product layer which requires a vector of fixed size.

- Resize data to expected input size for Network

- **Special Note**: if there are NO Fully connected layers, you do not need to do this. The output layer will be larger but, it does not represent classes but, feature vectors and can potentially be used in same way as before. One example Encoder-Decoder.
Option 2 – “Spatial Pyramid Pooling”

- "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition" by He et al. proposes a Spatial Pyramid Pooling layer.
- propose to add the Spatial Pyramid Pooling Layer just before the first fully-connected layer (details in the paper).

- hierarchically partitions the feature maps of the last convolutional layer (or the subsequent pooling or response normalization layer) into a fixed number of bins.
- Within these bins, responses are pooled as usually, creating a fixed-sized output.

#### Diagram

- Fully-connected layers ($f_{c_1}, f_{c_2}$)

- Fixed-length representation

- Spatial pyramid pooling layer

- Feature maps of conv3 (arbitrary size)

- Convolutional layers

- Input image
Multimodal DL: Dimensionality / Complexity of Network

- Multimodal sources drives up complexity of network – too large to train efficiently and data needs so great to achieve good performing systems

**GUIDELINE 7**: a Solution – Attention Processing

- Concept from Human Visual System: as our eyes scan and focus on one portion of image at a time. Consider reading, you look at one word/character at a time and scan to next. Your focus or attention changes. This reduces the amount of data you need to process at one time

- In Images this has a counterpart in the concept of ROI and localization.

Multimodal Attention Mechanism

- Mid-level DL network task, to determine focus on what “parts” of the currently processed mid-level features should be focused on.

Lots to explore – 1 mechanism for attention or different for each modality or some subsets? How can this be learned?

NOTE: fusion can take place earlier or later as shown here
“General” Concept of Attention Processing

- Mid-level in the DL network, after you have processed some features, it learns to “weight” some mid-level features more than others—which corresponds to only using portions of the original data that yielded those mid-level features.
- Here is an example of image captioning where parts of the image yielded mid-level features associated with a word in the resulting final DL output (see frisbee and corresponding attention area).

from “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, K. Xu, et.al. 2015

Conclusions

- Follow Guidelines
- “areas to explore”
  - Attention Mechanisms
  - Generative Networks
  - Pretraining and Retraining
  - Hierarchical multi-level fusion schemes
Multimodal Deep Learning: Issues and Solutions for Creating Well Performing Systems

Lynne Grewea and Chengzhi Hua

aComputer Science, California State University East Bay, 25800 Carlos Bee Boulevard, Hayward, CA USA, 94542

ABSTRACT

Multimodal Deep Learning is the application of Deep Learning Techniques with different sources and modalities of data. Examples of different modalities include text, images, video, audio as well as the myriad kinds of sensor data available. Multimodal Deep Learning adds to the complexity of the Deep Learning Task and create the specific issues of how to correlate/align multimodal data, how to fuse multimodal data, how to handle lack of/missing multimodal data, and how to improve performance with increased complexity of multimodal processing. Solutions and Guidelines to address these issues are discussed including correlation techniques, network architecture multimodal influences, the use of generative networks, autoencoders and multimodal transfer learning.

Keywords: Emotion Detection, Assisted Learning, Deep Learning, Vision, NLP

1. INTRODUCTION AND MULTIMODAL EXAMPLES

Machine Learning (ML) can be defined as a subset of Artificial Intelligence where mathematical models are applied to training data to learn systems (networks) that can perform the recognition task. No explicit programming of procedures takes place. Neural Networks including Deep Learning techniques, Support Vector Machines and Bayesian Networks are all examples of different models that would be classified as Machine Learning. What they all have in common is that the machines learn a model from training data. Multimodal Machine Learning implies that more than one kind of data or data from distinct sources are used. Humans use multimodal data sources continuously to make decisions. Our senses of sight, hearing, touch, and taste are used to help us navigate and understand our world. Taking this queue, it reasons that using multimodal data in an ML system could result in improved understanding. There are many issues related to the specific task of Multimodal ML including: data representation, correlation, feature learning, and fusion staging between modalities in ML.

In [1] a survey of multi-modal machine learning over the years is discussed focusing primarily on speech and vision. A review of Speech Recognition supplemented by vision, Indexing of Multimedia content by analysis as well as Media Description (e.g. video/image captioning) and Human Social Interaction (e.g. emotion recognition) applications are discussed. The applications for multi-modal ML is vast and include transportation, security, IoT, and exploration. There are too many papers and systems to present but, as an example the reader is referred to [7] for a paper that reviews Multimodal ML systems for Affect/Emotion Detection.

This paper focuses on the main challenges presented when multiple modalities of data are used in a Deep Learning System. First, a discussion of how to adequately describe and align or correlate different modality data is discussed. Secondly, the options for when fusion of the different modality data in a Deep Learning System are presented. The effect of having multiple modality data and the increased dimensionality of the data is discussed.
2. MULTIPLE MODALITIES – HOW TO DESCRIBE AND CORRELATE DATA

One important consideration, an what may be considered the first stage of any multi-modal ML system is how to represent the data and how to correlate the data between the modalities. In some ways, this may seem at the heart of the fusion ML question.

In [3], they identify desirable properties of multimodal data representations should include:

- similarity in the representation space should reflect the similarity of the corresponding concepts,
- the representation should be easy to obtain even in the absence of some modalities,
- it should be possible to fill-in missing modalities given the observed ones.

how to deal with different levels of noise; and how to deal with missing data.

- Taking direction from the long studied topic of stereo vision and 3D data calculation - the way to handle missing data in one modality is: to ignore it, to simply say it is not there. This usually means giving it some kind of null or default zero value. Another way to deal with missing data, if it is only minimal is to “borrow” from the surrounding data and interpolate the values. This is a potentially dangerous task as you are introducing new data that is not there and the questions of how much data you can do this for and how to interpolate are difficult to answer. With regards to handling noise that has different properties, in some ways the nature of modern Network based (like CNN) ML models through its learned features levels (convolution), noise can be “reduced” automatically and the spatial and temporal relational properties at the input layers for each modality of data are retained. Meaning you do not take one pixel of a Infrared image and say the next pixel in the image is from a 3D camera; instead you keep grouped in their raw data representation each modalities data and present this to the system for feature learning.

Another interesting idea is that Multiple Sources of information are not limited to coming from different sensors, modalities, but can be generated from a single modality. For example, in [9], a paper that creates 3 different sources of information from a single 2D image is described for use in a Deep Learning System to determine Human Pose.

Time Correlation and Location

With regards to representing the data, oftentimes the sensor and some form of digitizer define the representation of the data signal. For example, in video, the visual information is a time sequence of digital images at a fixed resolution. Corresponding to this is a sampled time sequence of audio. In other cases, for example the placement of multiple cameras in a security application in an airport, again the representation of the data is a time stamped sequence of images. If the clocks of the digitizers are aligned than, the different images from different cameras can be correlated by time. This is true also of any set of disparate, non-heterogenous sensors. In [15], a discussion of synchronization of clocks from different sensors is discussed.
Another metric that can be used which is a gross measure of correlation/alignment is location. Knowing the location for example with GPS coordinates is another way to roughly correlate or align multi-sensor data. However, this is crude and is better at rejecting samples as being aligned then actually aligning them.

**Spatial Correlation**

Often, we want to go beyond the simple correlation from time but, extend this more detailed correlations. For example, consider a visual spectrum image of 1000x1000 and you have a separate infrared sensor of 300x300 that is viewing the scene at a different angle. In this case, the two images would need to go through a rectification process (much like is done in stereo vision systems) where the transformation from one image space to another is performed via a calibration process. Of course, if the angles are too disparate, there is no or very little overlapping observance of the scene between sensors and then only a time stamp correlation can be performed. [16] discusses the use of a registration process for multiple sensors (much like stereo registration). Inertial information (geometric alignment) can also be useful in sensor data alignment (specifically here referred to as rectification).

![Fig M: Registration of IR and Visible spectrum images of same scene](image)

There are some publicly available multimodal datasets. For a review of the corpus of Audio-Visual data sets the reader is referred to [5]. Figure X shows a sample Audio Visual data correlation sample from CUAVE [6] where a sequence of audio is matched with a series of images from the corresponding video.

![Figure 1: CUAVE data sample from the Audio-Visual dataset](image)
When a “map” of the environment is known, it is possible to align different sensors data by first matching them to a pre-known model of the environment. Consider a mobile distributed sensor system (even humans walking around with mobile phones), and the multi-sensor stream of data coming into the system. This data could be aligned relative to each other via their determined “location” in the mapped environment. If two sensor’s data is determined to “overlap” or be in the same vicinity in the environment, then more detailed alignment could take place.

**Manual Similarity Function based Correlation**

Sometimes the data is not sensor driven but, can be human generated as shown in [11] where a system is discussed to align human generated text plot synopses to videos for use in story-based retrieval. This is an interesting case as while there is knowledge of the start to end sequence for both the text and video, there is no given knowledge of the exact correlation in time or space. In this case, where there is no synchronization (time correlation) of the data, a similarity function between the different modalities must be created and this can be used to correlate/align pieces of data from the different modalities by creating an optimization function where the maximum matches yield the best correlation/alignment of different modalities’ data. There are different techniques that can be applied to solve the optimization problem including Dynamic Time-Warping [12]. In [11], the system creates a similarity function using each sentence of the synopsis and comparing it to the subtitles of the video looking for matches in character appearance and keyword matches. They unfortunate problem with an explicit similarity function as shown in [11] is that it is manually developed and will not generalize to another system with different modalities. Also, it is certainly true that the use of subtitles – text- from the video in matching to the text of the plot makes development of a similarity function easier as the two modalities are inherently the same kind of data, text in this case.

**Learning Correlation**

If the data is again of fundamentally different types – like images and text then there is no direct underlying physical/structural spatial alignment (unlike there would be with an image and an IR image of the same scene). Consider the case of text description and an image. In this case, what can be achieved is some association with regions in the image and the corresponding text. A solution beyond similarity functions is using Deep Learning Networks themselves for the task of correlation/alignment. However, the generation of aligned multimodal data sets are required for the training of such a network. These kinds of datasets are starting to emerge like the newer Google Refexp (G-Ref) dataset [13] that uses the MS COCO dataset where each image is annotated with a few sentences/phrases and correlated areas in the image are specified. In [13] a CNN is used to represent the image (portion of image) followed by a LSTM network for text generation and the dataset is used for training and testing. [14] extends this work by not only aligning an image region with corresponding text, but adds the capability of text that is a “referring expression” (the dog is to the left of the boy) and involves multiple regions in an image by extending the structure of the network to run multiple LSTM networks in parallel each taking as input multiple detected image regions from a CNN.

To be fair, these systems are focused on generation of text from Images and as such are not strictly for alignment but, rather are for generation. However, they can be used for alignment of text with image or video sequences. As you can see the bulk of the work here deals with the modalities of text/audio and video. As aligned multi-modal datasets become available, Deep Learning can be used to solve the alignment task and this should be explored further.

![Region CNN](image.png)

**Fig A1:** Using a CNN followed by an LSTM to learn correlations for Text and Image (not can be used for text generation/captioning) from [13] where a proposed region is passed into the CNN.
How Much Correlation is Necessary?

An interesting question is how much correlation is desired or necessary at the front end of the system? The use of Deep Learning for correlation was discussed above, but, what if we simply present streams of data and let the system learn. Temporal alignment is necessary as Deep Learning Networks require training data and labeled data would minimally need to be at the same moment of the same scene. If it is possible to do more elaborate spatial correlation with relatively little noise introduction, then it is the opinion of this author that any following ML analysis will improve from the front end correlation.

GUIDELINE Multimodal Correlation: correlation of multi modal data allows for integration in a single DL network earlier (early-fusion). If correlation is naturally possible, it should be performed as likely better results will be achieved. Temporal correlation of multimodal data is typically necessary.
3. WHERE TO DO FUSION

Another design decision for Multimodal ML systems, is how to perform feature learning. In the recent past, design driven feature extraction algorithms were developed like SIFT in the early 2000s [4]. Currently the trend is for learned features not designed algorithms. For example, in a Convolutional Neural Network, this is primarily represented by the convolutional layers that are learned through training.

In modern network based ML feature learning systems for multimodal systems the question is if all the different modalities are processed by one network or separate networks or if a layered network approach is taken. The viewpoint taken in [2] is that feature learning is singularly important for the performance of an ML system, so this question is important.

As discussed in [2], general advice to improve feature learning is to apply dimensionality reduction as a pre-processing layer. For example, Principle Component Analysis (PCA) could be used. Interestingly and converse to this advise is the fact that most ML systems do not have such a preprocessing layer.

![Figure Y: Three kinds of Deep Learning Fusion Systems: Early (at input data level), Mid (after early stage network layers), Late (at final decision layer).](image)

**Fusion at the entrance level – joint ML system [Early Fusion]**

The most common and simplest representation of the data is to concatenate the individual modality features. In [17], the task of Speech Recognition is performed by comparing different fusion options including Early and Mid-level as well as comparing them to only Audio Deep Learning when the modalities of audio and video are present. This is a specific and popular example of multimodal deep learning. In this paper, CNNs were used to create the different fusion variations. They found that when no noise is present, the single modality outperformed any fusion for speech recognition. Interestingly, when noise is present an audio alone DL network performed better than a Mid Fusion DL network. However, when noise was present they found that combining a Mid Fusion DL network of audio and vision with a audio only DL in a Late fusion scheme (a hierarchy of fusion) performed better than audio alone. This is too specific of an example to draw any generalizations but, when the goal is to perform recognition with a strong dependence on one of the modalities, like the case here of speech recognition strongly dependent on the audio modality, there needs to be a strong presence of that modality’s DL network. In more complex scenarios like scene understanding, there is not necessarily a natural strong dependence on one modality.
Fusion after feature learning – layered approach – multiple ML sub-systems at front, fused part-way, final joint subsystem [Mid Fusion]

In [10], a deep learning neural network is proposed where a first stage of feature extraction on the separate modalities of speech and images in a video take place and the resulting features are fed into the Neural Network. The NN fuses the different modality features at the beginning layers. See next figure as example.

Figure X: From [10], feature extraction takes place followed by fusion.

Fusion at the end - after feature learning and decision — multiple ML systems, fused at final/decision layer. [Final Fusion]

This kind of system only fuses at the decision layer. It means that if a modality is missing, the output of the classifier should still function. In the early and mid-level fusion systems, the system would typically fail if a modality was missing. In [23], a system to explore the correlation between brain EEGs and 2D stimulus images where the 2D image is processed with a CNN yielding a EEG describing and the EEG input is processed with a LSTM encoder. The input to the late fusion stage are the EEG description yielded from the Image CNN and the features extracted from the LSTM EEG encoder which is trained to yield a classification of the EEG input.

Data can determine Fusion Level

The choice of architecture of a DL network can influence what type of fusion is possible with the particular multimodal data you have. For example, an image can be run “naturally” through a CNN or a Restricted Boltzmann Machine (RBM) or a Autoencoder NN. However, user preferences on movies would not be run through a CNN, this does not make sense. So, if you have an image and user preferences on movies as your 2 pieces of multimodal data, you could not do early fusion on them if wanted to apply CNN to the image. You would be forced to do Mid-level or Final-level fusion. However, if you have data that is all similar, like 3D images and infrared and RGB images, then all levels of fusion are possible with a given architecture. An example of a mid-level fusion is shown...
in [22] where the data includes 3D depth data, 2D RGB image data and human skeleton pose data. The 3D and 2D image data are fused together in early stages via a CNN. However, the human skeleton pose data is first processed with a Gaussian-Bernoulli Deep Belief Network. Then the output of these two are fused together in the overall DL network for the task of gesture recognition.

GUIDELINE Fusion Level: There is no overall best choice of where (early, mid, late) fusion should take place. Limitations on how multimodal data correlates determine if early fusion is possible. The earlier the fusion, generally the deeper the overall network may need to be. Because of correlation issues (typically image and text do not directly correlate), mid-level and late-level fusion are more common in multimodal DL system.

4. ISSUES: COMPLEXITY, NOT ENOUGH DATA AND MISSING MODALITIES
Multimodal Increased Complexity → Necessitates Deeper DL Networks

Generally, as the dimensionality of the data increases, which is a natural effect of increasing the number of sources, deeper networks are needed [2, MY SPIE paper on recommendations]. As networks grow deeper, generally more training data is needed. So, this makes training of Multimodal ML systems more difficult and expensive. One solution is to separate the networks, as discussed in section 3 “Fusion at the end” and have each modality train its own DL network and fuse them only at the decision layer or last layer(s). This is the least expensive of the solutions. A mid-expense solution would be to layer in the networks so that, some form of feature learning is first performed separately for each modality and these output layers representing higher level features (more abstract information) are then fused together into a shallower feature network leading eventually to a (fully) connected output decision layer.

It could be argued that presenting the ML system with all the modalities of data at the same time, given enough training data, will result in the best solution as with a sufficiently deep network the combinations of data elements into learned features is greater. This is true if there is good correlation between the data. It is unclear what the results will be when the data is not directly correlated between the modalities, in that case it may be that doing fusion at the final or later decision layers could provide superior results. Only a direct comparison of different models will tell for any one understanding multimodal data problem. Deep architectures can lead to abstract representations because more abstract concepts can often be constructed in terms of less abstract ones.

GUIDELINE Network Depth: to reduce depth of DL network consider mid- or late- fusion architectures

Not enough Data → Solution 1: Transfer Learning

Transfer learning, or the retraining of an existing network, is a technique that is particularly useful when you do not have a lot of training data (or enough to adequately model). It has been shown [****], that transfer learning can help dramatically improve results in this case. However, the premise is that you have similar input data and similar output classes. If you do not, retraining may not help and could give poor results. The only way to know is to compare a retrained network with one that is trained from scratch. In [MY SPIE paper], I have given some guidelines to consider when to retrain. However, because there exist so little publicly available multi-modal ML systems, there are none to retrain. Also, because multi-modal systems are often unique in nature, and choice of their sensors, it means that the network with not be readily applicable for retraining. As discussed in [drone paper], it may be possible that you will have to significantly alter the network to add new modalities in to perform retraining. One idea is to create a new network where the beginning layers are separated by modality and the
Not enough Data → Solution 2: Pretraining with Autoencoders

When you don’t have a lot of labeled data, besides retraining an existing model, another possibility to assist with performance is pre-training via autoencoders via unsupervised learning. An autoencoder is when you present the unlabeled input and the output should be the same input labels (this is all you have with unsupervised learning). The middle layer of the CNN contains a set of nodes that represent a condensed feature-rich representation of the input data. The “pretrained” network up to the middle layer can be used as the front your deep learning network where you add some output layers for classification and then train this new network with some of your labeled training data set.

By training the autoencoder with a larger set of unlabeled data, in essence you are learning the feature level beginning layers of your network first. Then this network is “retrained”/ fine tuned along with the training of the new final layers using your limited labeled data. Note you can have straight forward NN Autoencoders or CNN based Autoencoders.

![AutoEncoder Diagram](image-url)

Figure Y: AutoEncoder basically learns to reconstruct the original input signal –here an image. The middle layer can be thought of as a condensed representation, feature-rich representation of the input. You can take the network up to that point and add final layers for classification and retrain with your limited set of training data samples.

| GUIDELINE | Lack of Training Data: Consider Retraining (transfer learning) & Autoencoders for generation. Today, most models available for retraining are single modal and this means the multimodal DL architecture can not involve early-fusion. |

Missing Modalities /Data → Solution: Generative Autoencoders

Related but, different than not enough data is the idea that only one of the modalities may lack data or in a run time environment at one moment in time not be present due to sensor malfunction. Autoencoders can be used as Generative networks and generate missing data for one modalities data from the other modality data that exists. A DL network called a Cascaded Residual Autoencoder, a stacked set of autoencoders, is used to infer missing
modality data in [24]. While many kinds of Autoencoder network architectures can be envisioned [25,26], Figure Z shows the concept.

In [8] a discussion of what happens when different modalities are present (or missing) at different learning stages of Feature Learning (i.e. convolutional layers in a CNN), Supervised Training and Testing. The idea is that all modalities are present for feature learning but, not for training and testing of the ML system. In one case, called Cross Modality Learning only one modality data is present for training and testing which tests if better ML system can be learned for one modality when features are learned using all modalities. The last case called Shared Representation Learning which uses a different modality for training than testing and this studies if the ML system has learned representations that are invariant of the modality. In [8], they show on two datasets that Cross Modality Learning is somewhat superior to classical single modal learning.

![Diagram of a Special Generative Autoencoder Network](image)

**Figure Z:** Special Generative Autoencoder Network trained to generate one modality data (here depicted as EEG images) from other modality(s) (here 2D RGB image).

**GUIDELINE Missing Modality Data:** Use Generative Autoencoders to generate missing modality data

---

5. **MULTIPLE MODALITIES & ACHIEVING BETTER PERFORMANCE ➔ FOCUSING ATTENTION**

Attention describes the process in the human-visual system where rather than processing at mid and later stages all data, the most salient features are selected/emphasized for further processing [18]. Much like our eyes move as we read, or how we move our eye focusing on different parts of a moving scene (first batter, then bat, then ball flying), the attention changes, what part of the data we examine or process changes. This concept seems to imply temporally changing data (like video), but, can be applied to a single time piece of data (like an image). In fact, in [18] the authors propose to use an attention processing stage in their Deep Learning Network to reduce the size of the data when working with normal resolution images and to perform localization of detected objects, in essence the attention mechanism here is performing a kind of ROI detection. The Attention process can be used to reduce what might be examined which can lead to better performance for the DL network both in terms of time to process as well as potentially better accuracies (looking only at what matters, ignoring the clutter).
Attention processing is more commonly used in temporal data sequences like video captioning. The concept in the application of video captioning is to focus the attention on only part of the incoming data and not all of it in the captioning task. This is typically done in a system trained end to end that includes a mid-level network stage performing attention processing by selecting which parts of the early processed data to continue to process on in the remaining stages of the network. For example, in [17], the first stage of the Deep Learning Network applied to the video captioning task, takes small sequences of video (image frames) and produces a set of descriptors (e.g. “man”, “balloon”, “cat”, “feeds”, “bird”) using a CNN and the second “stage” of the DL Network form an attention mechanism that selects the better descriptors from each small video sequence. This second stage attention mechanism basically applies weights to the small time sequence video descriptors coming from the first stage and these weights are learning through training. The weighted descriptors (e.g. “man”-0.9, “balloon”-0.1) are then fed into the final/3rd stage of the DL network which is implemented using a Long Short Term Memory Network (LSTM) to produce the final video sequence (e.g. “man feeds cat”).

Basically, the concept of an attention mechanism is a mid-level network process to reduce the data volume by focusing in on what is more likely to be important given the training data. This could be applied to general multimodal systems to not only emphasize what is most important in each modalities data but, also amongst the modalities what is important. As it is part of training an end-to-end network, it relies only on specifying good training data and some specification in the mid-level of a weighting mechanism where the weights are part of the learned network [18].

Introducing a mid-level attention mechanism in your DL network may also be a way of taking a pretrained network trained on a wide generic data set and creating a custom DL network for retraining where the later stage involves classification to your particular set of classes or in the case of temporal sequence networks (LSTM) a temporal sequence of class decisions. The idea is that the weights in the mid-level attention processing sub-network may be more quickly learned than just adding some final neural network layers as is done in a typical retraining scenario.

In the case of multimodal attention, a question to be answered is should the attention mechanism be the same and applied to all modalities at the same time as shown in Figure E or should there be separate mechanisms for each modality. In [20], the task of translation of video captions is performed using multimodal attention processing. Here the already matched image and corresponding Text are fed into the system and the output is “translation” / generation of captioning in a different language (or regeneration of same language). So, an input is an image and some text in English (e.g. woman in black dress riding a scooter down a street), and the output will be a text phrase in German or regeneration of the text in English (a woman in black dress on a scooter). In [18], they compare using the same learned attention process for both modalities or having separate learned attention processes for the two.
modalities. In this specific task, they found separate attention processes performed better. However, this can not be generalized to other kinds of modalities (like say 3D and 2D images or sound and images).

**GUIDELINE performance:** Consider an Attention process to focus computation to improve time and possibly accuracy performances.

6. REFERENCES

Talk in one Slide

- Use shape-based mathematics to help sensor-related tasks
- First vignette: topological data analysis (TDA) applied to Multiple-Hypothesis Tracking (MHT) for passive EO
- Second vignette: self-similarity, random walks, wavelets applied to audio-visual fusion in speech recognition
- Prelude: some words about the use of shape analytics in ML
Collaborators, References, Funders

- **Multi-scale Geometric Summaries for Similarity-based Sensor Fusion.** Christopher J. Tralie, Paul Bendich, and John Harer. submitted to 2019 IEEE Aerospace Conference
- Also includes work by: Lihan Yao, Alex Pieloch, Steve Marron, Ezra Miller, Sean Skwerer, Nathan Borggren, Ken Stewart
- Funders: OSD, AFOSR, AFRL, NSF.

---

(Supervised) Machine-Learning

- Pairs of training data \((x_1, y_1), \ldots, (x_n, y_n)\)
- Thought of as many draws from random variable pair \((X, Y)\)
- Discrete \(Y\): classification; Continuous \(Y\): regression
- Most algorithms want \(X\) to be vector-valued, or at least want a similarity metric of some kind.
Shape and ML

- ML gets interesting when $X$ has more shape-based structure.
- (Motta, 2018) good survey on topological feature extraction for ML.
- (Zhang et al 2018) and (Gabrielson and Carlsson 2018) use geometry and topology to understand deep networks.

First Vignette
### Tracking: some basic problems

- Objects moving/emit, observed by sensors
- Common sensor modalities: radar, EO/IR, sonar, . . .
- Sensors scan, make measurements and reports: target of interest, clutter, scene background, . . .
- Tracking task: take data and partition it into tracks corresponding to targets
- Hard already for single target, harder still for multiple targets

### Target Behavior into MHT via TDA

- **Big idea:**
  - Use “similarity of behavior” to connect-the-dots
  - TDA: use PDs of motion primitive signals to assess “behavior”
  - Produce behavior likelihoods that get integrated into MHT
- **Proof-of-principle:**
  - Experiments with well-respected synthetic data
  - Drastically improved data association results, especially in noisier sensing contexts
Simulated Data: SUMO

- Simulated Urban MOBility (sumo.sourceforge.net); open-source, developed by Institute of Transportation Systems at the German Aerospace Center
- Allows user to:
  - import road grid
  - simulate a variety of driver characteristics (lane-switching tendency, reaction to stop lights, following distance, turning radius, ...)
  - add code to simulate output of various sensors

Experimental Setup

- Took a road grid of an American city
- Let 1000 vehicles drive around
- Two driver-behavior classes: “aggressive” and “normal”
- Chopped tracks up into many (~ 100,000) “tracklets” of length N
- SL task: extract features from motion-primitive signal snippets taken from tracklets to infer behavior class
Zero-dimensional Persistence Diagrams

- Signal snippet $f : [a, b] \rightarrow \mathbb{R}$
- Threshold set $X_c = \{x \in [a, b] \mid f(x) \leq c\}$
- PD tracks evolution of components of $X_c$ as $c$ increases
- Pair threshold values which create components with those that merge them; also pair global min and global max values.

Paul Bendich (Department of Math, Topology, Geometry, and Machine Learning)

Proc. of SPIE Vol. 11018  1101801-88

Terms of Use: https://www.spiedigitallibrary.org/terms-of-use
Zero-dimensional Persistence Diagrams

- Signal snippet \( f : [a, b] \rightarrow R \)
- Threshold set \( X_c = \{ x \in [a, b] \mid f(x) \leq c \} \)
- PD tracks evolution of components of \( X_c \) as \( c \) increases
- Pair threshold values which create components with those that merge them; also pair global min and global max values.
Zero-dimensional Persistence Diagrams

- Signal snippet \( f : [a, b] \to R \)
- Threshold set \( X_c = \{ x \in [a, b] \mid f(x) \leq c \} \)
- PD tracks evolution of components of \( X_c \) as \( c \) increases
- Pair threshold values which create components with those that merge them; also pair global min and global max values.

PDs for Speed Profiles
Supervised Learning

- Trained both logistic regression and Poisson model (5-fold CV).
- Good classification results, even for small snippet size; much better than simple “max speed” classifier.
- Learned Poisson rates (log scale) show subtle class differences: e.g., aggressive drivers tend to have more low/medium persistence wobbles at higher speeds.

TDA-augmented MHT

- Trained models lead to behavior likelihoods
- Behavior likelihoods integrated into new MHT
- Lots and lots of hard details here!
  - multi-rate: behavior likelihoods updated over larger intervals than more primitive kinematic-based likelihoods
Improving Tracking with TDA

- Proof-of-principle experiment
- SUMO simulation of two drivers approaching intersection, stopping, then continuing
- Simulated passive EO reports angle-only measurements, with increasing uncertainty
- For each uncertainty level:
  - run 100 trials for both standard and TDA-enriched MHT
  - report whether correct data association occurs

Second Vignette
OuluVS2 Digits Dataset

- Speakers utter 10 digit strings (e.g., 1, 2, 5, 5, 7, 8, 8, 1, 3, 0)
- 51 speakers
- 10 sequences, 3 instances per speaker per sequence
- Video from multiple points of view, audio


Problems and Success Metrics

- Decompose set of digit strings various ways:
  - by digit string, by speaker, by speaker and digit string
- Goal is to come up with similarity ranking mechanism $\mu$ so that:
  - For each object $s$, $\mu(s, t)$ is larger when $t$ is in same class as $s$

(Rusinkiewicz and Funkhouser, 2009)
Problems and Success Metrics

- Success evaluated by precision-recall curves for each object $s$
- Report average P-R curves
- Area under P-R curve is *mean average precision* (MAP)

**Other approaches, our pipeline(s)**

- Many approaches (including ours) construct $\mu$ via mapping strings into a feature space.
- Lots of deep-learning approaches (Lopez and Sukno, 2018: survey)
- One very cool paper (Sargin et al, 2007)
  - Train HMMs per class on audio data
  - Use Canonical Correlation Analysis to learn good ways to extract fused audio/visual features
- We propose a set of entirely *unsupervised* pipelines (labeled examples used only to *evaluate*, not to *train*)
Self-Similarity Matrices (SSMs)

- Given curve $\gamma : [a, b] \rightarrow (M, \rho)$, define SSI $D : [a, b] \times [a, b] \rightarrow \mathbb{R}$ via $D(s, t) = \rho(\gamma(s), \gamma(s))$.
- Discretize $a = t_1 < t_1 < \ldots < t_n = b$, form $n \times n$ SSM $D_{ij} = \rho(\gamma(t_i), \gamma(t_j))$.
- Can transform into weight matrices $W$ where 1 is close and 0 is far.

Why SSMs?

(Junejo et al, View-independent action recognition from temporal self-similarities, PAMI, 2011)
Pre-processing:

- **Video:**
  - extract lip region from each frame and rescale to $25 \times 25$ grayscale
  - Treat as time series in $25 \cdot 25 = 625$ dimensional Euclidean space.

- **Audio:**
  - Break audio signal into overlapping windows
  - Summarize each window via 20 mel-frequency cepstral coefficients (MFCCs)
  - Standard technique in audio signal processing
  - Treat as time series in 20 dimensional Euclidean space

(Bogert et al, *The quefrency analysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking*, 1963.)

SSMs/MFCC in Cover Songs

- SSMs of MFCC time series previously used in automatic cover song recognition
- Previous conventional wisdom: notes (chroma) best for this problem
- turns out: MFCC (timbre) plus geometry has a lot to say

(Tralie and Bendich, *Cover Song Identification with Timbral Shape Sequences*, ISMIR 2015)
Similarity Network Fusion (SNF)

- Transform several weight matrices $W_1, \ldots, W_m$ into one that (hopefully) has best qualities of all.
- Technique based on random walks with cross-talk between matrices for probabilities.

(Wang et al, *Similarity network fusion for aggregating data types on a genomic scale*, Nature Methods, 2014)

---

SNF for early audio-visual fusion

- SNF used (Tralie, ISMIR 2017) to fuse Chroma and MFCC on cover song problem, current state-of-the-art
- Here we use it to fuse MFCC (audio) and lip pixel (video) SSMs
How to Compare SSMs?

- Each string $s$ transformed into SSMs $WA(s)$, $WV(s)$, then fused into $WF(s)$.
- How to compare $WF(s)$ with $WF(s^t)$? could just use $\ell_2$ (Matrix Frobenius Norm)

Local delays (time warps) induce local perturbations in SSMs
$\ell_2$ norm unstable to these perturbations
How to Compare SSMs?

- Each string $s$ transformed into SSMs $W_A(s)$, $W_V(s)$, then fused into $W_F(s)$.
- How to compare $W_F(s)$ with $W_F(s^T)$? Could just use $\| \cdot \|_2$ (Matrix Frobenius Norm)
- We extract features from SSMs using the scattering transform, with nice theoretical stability properties.

![Image](https://example.com/image.png)

SNF for late audio-visual fusion

- Everything so far has happened upstream: before ranking decisions are made.
- Can also apply SNF downstream.
- Given object-level metrics $\mu_1, \ldots, \mu_k$ on set of $N$ objects (strings).
- Each one produces object-level SSMs, which can themselves be fused into a new SSM.
- We apply that here with $k = 3$. 

![Diagram](https://example.com/diagram.png)
**Results: Digit String Identification**

![Graph showing precision recall curves for digit sequence, no noise.]

- Video2 (0.386)
- Audio2 (0.323)
- AllLateFused2 (0.905)
- EarlyFused2 (0.433)
- AVLateFusedScatter (0.978)
- VideoScatter (0.676)
- AllLateFusedScatter (0.982)
- AudioScatter (0.731)
- Random (0.104)
- EarlyFusedScatter (0.756)

---

**Results: Digit String Identification, with simulated noise**

![Graph showing varying noise mean average precision by digit sequence.]

- Video2
- Audio2
- EarlyFused2
- AllLateFused2
- VideoScatter
- AudioScatter
- EarlyFusedScatter
- AllLateFusedScatter
- AVLateFusedScatter
- Random
Results: Speaker Identification, with simulated noise

![Varying Noise Mean Average Precision by Speaker](image)

Results: String and Speaker Identification, with simulated noise

![Varying Noise Mean Average Precision by Digit Sequence And Speaker](image)
THANK YOU FOR YOUR ATTENTION
Topology and Geometry for Tracking and Sensor Fusion

Paul Bendich\textsuperscript{a,b}

\textsuperscript{a}Department of Mathematics, Duke University, Durham, NC, USA
\textsuperscript{b}Geometric Data Analytics, Durham, NC, USA

ABSTRACT

This position paper surveys a recent flurry of research activity which incorporates shape-based mathematics (topology and geometry) into machine-learning (ML). Particular focus is given to two recent papers, one which integrates topological data analysis (TDA) and supervised learning within multiple hypothesis tracking, and one which uses a variety of multi-scale geometric techniques to aid in unsupervised heterogeneous sensor fusion.

Keywords: Tracking, sensor fusion, machine-learning, topology, geometry, multiple-hypothesis tracking, self-similarity, wavelets

1. INTRODUCTION

Topological data analysis (TDA \textsuperscript{10}) has been around for almost twenty years, and has helped to build a bridge\textsuperscript{7} between abstract areas of algebraic topology and data-driven applications. This position paper describes (Section 2) some interactions between TDA, along with other multi-scale methods of extracting features using shape, and machine-learning (ML). Particular focus is given to works incorporating (Section 3) TDA into vehicle-tracking, and to multi-scale geometric methods (Section 4) for heterogeneous sensor fusion. A key thread throughout is an emphasis on how multi-scale shape-based methods can transform seemingly disparate objects into a common arena for analysis.

2. SHAPE ANALYSIS AND MACHINE LEARNING

A typical setup for a supervised learning problem is to start with many labeled pairs \((x_1, y_1), \ldots, (x_n, y_m)\), each a draw from idealized random variables \(X\) and \(Y\) drawn on the same outcome in a sample space. The point of any supervised-learning algorithm is then to use the given pairs to make inferences about the value of \(y\) for any new value of \(x\). Many ML algorithms make the implicit assumption that the objects \(x_i\) are Euclidean vectors. For example, multiple linear regression models \(Y \approx \beta^T X\). Less restrictively, a classification method like \(k\)-nearest-neighbor requires at least the existence of a natural distance measure between the \(x_i\).

There are, however, many interesting situations where the data objects\textsuperscript{18} \(x_i\) do not initially admit such a representation. For example, the data objects in one work\textsuperscript{5} are tree-like arterial structures in the human brain, and one is attempting to do regression against age. Another paper\textsuperscript{1} considers data objects which are proposed reconstructions of road networks from GPS data and classifies them according to the original network that generated the data. Here a sensible thing to do is to first transform each data object into a feature vector and/or construct a reasonable distance measure between them, in a way that somehow takes account of the multi-scale shape of the object, and then use more standard supervised-learning algorithms on the result. Motta\textsuperscript{19} has a nice extensive survey of many shape-based feature extraction papers. Sections 3 and 4 give a deeper dive into two papers involving the author.

In addition to the above works that use shape-based features as a front-end to standard ML pipelines, there has been a recent surge of activity that uses shape to provide insight about what is happening inside deep network pipelines. For example, Zhang and collaborators show\textsuperscript{27} that certain feedforward networks can be understood using principles from tropical geometry, while Gabrielsson and Carlson use\textsuperscript{12} TDA to characterize properties of convolutional deep networks.

Further author information: (Send correspondence to Paul Bendich)
bendich@math.duke.edu
3. TOPOLOGY AND TRACKING

The target-tracking task is to partition sensor data into tracks that are produced by the same target. A key challenge is to take a sensor observation at a given time and associate it with a previously-existing (or new) track. Many, but not all, high-level designs for multi-target multi-sensor tracking systems fit within the multiple hypothesis (MHT) paradigm. Typical MHTs formulate the ‘connect-the-dots’ problem as one of Bayesian inference, with competing multi-track hypotheses receiving scores, for example via the Bayesian Log-Likelihood Ratio. A deferred decision logic scheme is used to process observed data in recursive fashion, aiming to keep the list of candidate tracklets (alternative attempts to represent targets with data) not too much larger than the actual number of targets.

This section describes a recent effort which uses TDA to produce behavior-based likelihoods whose integration into an MHT dramatically improves performance on a principled synthetic scenario. More mathematical details appear in Bendich et al and more tracker-specific details can be found in Rouse et al. There are a few other papers (e.g. Hossny et al) which also use topological methods to quantify driver behavior, and one that performs topological simplification of trajectories. There are two main high-level insights here. First, having a quick way to assess driver for a high-confidence tracklet can enable hypothesis-pruning via behavior-matching. As a simple version of driver behavior, many tracklets were simulated using the SUMO engine, partitioned into normal and aggressive driver classes. The first and third panels from left in Figure 1 show speed signals for tracklets of drivers in these two classes. Second, a TDA technique called the persistence diagram behaves extremely fast and provably robust mechanism for extracting parsimonious features from signal snippets. The second and fourth panels of the same figure show the persistence diagrams extracted from the snippets. There are many other ways to use persistence diagrams in signal analysis; e.g. via sliding-window embeddings to detect chatter in dynamical systems. A simple SVM classifier was then used to almost perfectly distinguish these behavior classes.

Likelihoods derived from the SVM classifier were then integrated into a multi-rate MHT, whose benefits were illustrated in a proof-of-principle experiment. The leftmost panel of Figure 2 shows a normal driver and an aggressive driver approaching an intersection, while the middle two panels show different tracker performances after the intersection. The TDA-enriched tracker consistently outperformed the original one, as measured by proportion of correct association plotted against simulated noise, and the performance delta got better under increasing levels of simulated sensor noise.
4. MULTI-SCALE GEOMETRY AND SENSOR FUSION

Many sensor systems involve sensors from multiple modalities. Thus different fusion techniques are required to perform tasks such as tracking or classification. This section surveys a recent extremely general and unsupervised fusion technique; full mathematical details appear in this paper. One imagines several behavior sequences observed by a heterogeneous sensor system. Domain-specific pre-processing leads to a time series (in any dimension) associated to each sensor modality. The multi-scale geometric shapes of these time series are abstracted from their (potentially distinct) ambient dimension and represented in a common space via two dimensional self-similarity matrices (SSMs). The left panel of Figure 3 shows three notional examples. SSMs have been used in many applications, such as activity recognition. The modality-specific information is then fused at the SSM level via similarity network fusion (SNF), a diffusion-based technique which produces fused SSMs that accentuate the best properties of the stovepiped ones, as illustrated in the middle of the same figure; SNF has been used previously for automatic cover-song recognition. SSMs are images and hence already live in a pixelated vector space. But small time warps in a time series induce drastic changes in SSM, as illustrated in the right of the same figure, and so a more robust featurization choice uses the wavelet-based scattering transform.

The benefits of this pipeline were first demonstrated in the audio-visual fusion context on a very well-studied dataset. This dataset consists of 10-digit strings uttered a large number of times by a variety of speakers while under observation by both microphones and cameras. Domain-specific pre-processing techniques were then used: each audio track was turned into a time series in \( \mathbb{R}^{20} \) using MFCC, and the lip regions were extracted from video and uniformly re-sized to form a time series in \( \mathbb{R}^{625} \). These time series were fed through the above fusion pipeline to produce a vector scattering coefficients. Simple Euclidean comparisons between these vectors then served as a classifier which achieved near-perfect performance on the string recognition tasks. We note that most previous attacks on this problem use extensive supervision and deep network techniques, whereas the approach surveyed here uses labeled examples only for evaluation, not for the method itself. The pipeline is also currently being used in the seismic-acoustic fusion context, with ongoing work that will demonstrate its benefits using the recently-released ESCAPE multi-modal dataset.

ACKNOWLEDGMENTS

At various times during this effort, the author has been partially supported by the Air Force Office of Scientific Research under grant AFOSR FA8750-17-C-0054, the Air Force Research Laboratory ASIMOV program under AFRL-RIKD Contract FA8750-18-C-0009, the Office of the Secretary of Defense under the OASIS JHU Sub-contract 110279 on HQ0034-12-C-0024, as well as the National Science Foundation under the BIGDATA grant, DMS 144749.

REFERENCES


MaxEnt DDF for Enhanced Detection and Implicit Context Detection

- **Motivation:** An information-theoretic-metric based, adaptive distributed data fusion (DDF) system with the ability to learn the performance of sensors and associated context, utilizing unknown prior probabilities-based local binary decisions which are sent to a parallel architecture fusion center producing global decisions.
- **Approach:** Based on the concept of machine learning, a maximum entropy-based (MaxEnt) parametric model is used for supervised classification and prediction, and serves as the global optimum linearly weighted decisions rule (sufficient statistics).
  - The optimum weights are learned without requiring conditional independence of local decisions.
  - The system requires an initial set of random training data after which it learns actual sensor detection performances, and recognizes context induced changes (e.g., contested environment, targets hiding, change of sensors, sensor failures, etc.).
  - Simulation results show MaxEnt DDF achieves better detection performance than Bayesian and Neyman-Pearson detection criteria based models.
Distributed Decisions Fusion (DDF) for Enhanced Detection

**MaxEnt Model:** Why use MaxEnt-based classifier?
- MaxEnt means minimum commitment by choosing the most “uniform” distribution, i.e., one with maximum entropy.
- Model all that is known (sensor decisions are modeled as feature vectors (FVs)) and assume nothing about what is unknown.
- The known FVs have to satisfy a set of constraints which is solved by simple convex optimization method as part of estimating optimum fusion weights.

- **Simulation Example: Bayesian Case**
  - No ad-hoc fusion rules needed.

---

**About Maximum Entropy (MaxEnt)**

- MaxEnt has been used for parameter classification and prediction when we don’t know anything about the actual prior probabilities, but we can model the underlying process.
- Why use Maximum entropy-based classifier?
- In real-world applications sensor performance parameters may not be known.
- Maximum Entropy $\Rightarrow$ Minimize commitment, choose the most “uniform” distribution, i.e., one with maximum entropy.
- Model probability estimates should reflect what we know and what we don’t know: “in general, ignorance is preferable to error”.
- MaxEnt has been applied to several areas, e.g., word disambiguation it text search, image processing/reconstruction, distributed detection, fault detection, sensor measurements fusion, econometrics, etc. Evolved by correspondence of entropy in statistical mechanics [Jaynes 1957] and Shannon entropy.
The General MaxEnt Framework

• The first step is to construct a statistical model which represents the behavior of a process by collecting a set of training data from it, e.g., in document text search use contextual info x, to find an output y.

• That is, obtain samples \((x_i, y_i)\) where \(x_i\) represent contextual information about words (e.g., “respect”) from the document, and \(y_i\) its class (“noun”). This yields an empirical probability distribution, where \(N\) is the size of the training data set:

\[
\hat{p}(x, y) = \frac{1}{N} \text{times}(x, y) \text{ occurs in sample}
\]

• RE: Above, one can introduce a function, say an \(N\)-dimensional real-valued feature vector function, \(f(x, y)\); one feature can be an indicator function:

\[
f_i(x, y) = \begin{cases} 
1 & \text{if } x \text{ is a 'respect' and } y \text{ is Noun; } \ 0 & \text{otherwise}
\end{cases}
\]

• In application of MaxEnt to DF, the feature functions represent binary declarations from sensors (which are results of hypothesis tests), and depict presence (detected) or absence (not detected) of a target.

• The initial training data for target detection can be generated from a uniform distribution \([0, 1]\), target present or absent (i.e., detected or not detected) is equally likely.

The General MaxEnt Framework

• The expected value of \(f(x, y)\) with respect to the empirical distribution \(\hat{p}(x, y)\) is the statistics needed next (note: in notation below will use \(f(x)\) for feature vector and \(\hat{p}(x)\) for the empirical distribution): viz., the expected value of the feature vector from the observed probability of \(x\) in the training data:

\[
\hat{E}[f] = \sum_{x \in X} \hat{p}(x) f(x)
\]

• The goal is to construct a model distribution \(p\), which satisfies the constraints imposed by the empirical distribution \(\hat{p}(x)\), viz.,

\[
E[f] = \hat{E}[f] \tag{1}
\]

• i.e., the expected value needs to be equal to empirical training value, and guarantee that our model \(p\) is proper probability distribution,

\[
\sum_{x \in X} p(x) = 1 \tag{2}
\]

• That is, the classifier is to learn from the training data, and use it to classify (match) incoming information and predict outcome. In general, the training data could also consists of collected random historical data from a particular DF system.
The General MaxEnt Framework

• The stated problem is ill-posed, and many models would satisfy Equations (1) and (2).
• The most appropriate is to select Jaynes' (1957) Principle of Maximum Entropy: “in the absence of additional information, assume that all events have equal probability”.
• That is, assign the highest prior probability to distributions which maximizes Shannon’s information entropy:

\[ H(p) = - \sum_{x \in X} p(x) \log p(x) \]

• The above is constrained optimization problem, which is solved by Lagrange multipliers, with result yielding the parametric form of MaxEnt exponential classifier distribution shown below:

\[ p(x) = \frac{\exp(w^T f(x))}{\sum_{y \in X} \exp(w^T f(x))} \]  

where, \( w \) is an N-dimensional parameter vector, and \( w^T f(x) \), which is the inner product of parameter vector and the feature vector.

The General MaxEnt Framework

• The information theoretic justification of Jaynes was generalized by Kullback (1959) by showing that the measure of closeness is the relative entropy, known as the Kullback-Leibler divergence.
• That is, when \( p(x) \) is uniform the minimum relative entropy criterion is the same as the maximum entropy criterion, which is used below.
• Given the model in (3) the next step is constructing a MaxEnt distribution to find the values for \( w_i \) such that

\[ \sum_x p(x) f_i(x) = \hat{E}[f_i(x)] \]

• To achieve the goal its been suggested by Berger (1996) to consider the use of the conditional probability of \( p(x) \), i.e., given a set of contexts computing the conditional probability \( p(x) \), where an event \( x \) in context \( c \), in general sets, is:

\[ p(x \mid c) = \frac{\exp(w^T f(x))}{\sum_{y \in G(c)} \exp(w^T f(y))} \]
The General MaxEnt Framework

• Given (4), fitting a MaxEnt model to a collection of training data requires finding the values for the parameter vector, which minimizes the Kullback-Leibler divergence between the model $p_w$ and the empirical distribution $\hat{p}$.

$$\text{Div}(\hat{p}, c) = \sum_{i,x} \hat{p}(x,c) \log \frac{\hat{p}(x,c)}{p_w(x / c)}$$

which, can be shown equivalent to maximize log-likelihood (Berger 1996):

$$L(x) = \sum_{c,t} \hat{p}(x,c) \log p_w(x / c)$$

(5)

• Here one needs to find the maximum of a concave function, and use Lagrange multipliers to reach a solution.

• Given the above overview of MaxEnt, the following slides will show MaxEnt application to DF, including solution of (5).

Distributed Decision Fusion (DF) - Parallel Topology

Note: *Bayesian or N-P (standard DF methods) **MaxEnt Machine Learning

For standard DF j is not applicable.
MaxEnt Application to DF

Given the general description of MaxEnt, the end goal of MaxEnt used for classification, using indicator feature functions \( f_i(x, y) \) is to compute the global prediction \( \hat{U} \) by the model, which maximizes the conditional probability, (as shown in subsequent slides), in reference to the parameters defined in Figure DF, and noted below:

- \( S_i \) is the \( i \)th sensor. Based on the local detection criteria and threshold at each sensor, it makes a binary decision \( u_j \), where, \( i = 1, \ldots, N \); and \( j \) is the system time when decisions are made or iteration time/sample numbers in simulation. Note: the values of \( u_j \) are either 0 or 1.

- The binary decisions are assumed sent synchronized to the fusion center for decision fusion.

- The approach doesn’t assume knowledge of actual prior probabilities, and is not dependent on individual sensor detection scheme.

- The optimum weights can be estimated without requiring conditional independence of local decisions.

Using the binary decisions variables notation, and substituting the feature functions into Equation (4), the resultant expression for the conditional probability is stated in Equation (6)

\[
P(u_j | u_{j1}, u_{j2}, \ldots, u_{jN}) = \frac{\exp\left[\sum_{n=1}^{N} \omega_n f_n(u_{j1}^{n}, \ldots, u_{jN}^{n}, u_j)\right]}{\sum_{x=0}^{1} \exp\left[\sum_{n=1}^{N} \omega_n f_n(u_{j1}^{n}, \ldots, u_{jN}^{n}, u_j = x)\right]} \tag{6}
\]

where, \( f_n(u_{j1}^{n}, \ldots, u_{jN}^{n}, u_j) \) is the \( n \)th feature function, \( \omega_n \) is the weight parameter of the \( n \)th feature function to be estimated, and \( x \) is either zero or one (thus the denominator in (6) is a sum of two terms) indicating the binary declarations of sensors.

- The \( n \)th feature function can be further specified as:
  \[
f_n(u_{j1}^{n}, \ldots, u_{jN}^{n}, u_j = x) = \begin{cases} 1, & \text{if } u_{j1}^{n} = y; \text{and}\ldots; u_{jN}^{n} = k; \text{or}\omega_n = 0 \end{cases}
\]

- where, \( n \) is the \( n \)th sensor, and \( k \) corresponds to the sensor declaration state 0 or 1.
MaxEnt Application to DF

- In order to estimate $\mathbf{w}$, we need to solve (5), which can be restated in the DF problem framework:
  \[ L = \sum_{j=1}^{DT} \log P(u_j / u_j^1, u_j^2, ..., u_j^N) \]
  - where, the sum is over the number of training samples, $DT$, which equals the number of samples in the data set. 
  - Since $L$ is a concave function a global maximum can be found by standard convex optimization methods, such as gradient ascent.
  - Used gradient ascent. The solution yields the optimum weights once the algorithm converges, $n=1, ..., N$.
  - Simulations show that the algorithm converges within fewer than fifty iterations. Note that this algorithm is executed for each data sample processed.
  - Having learned the optimum weights, we can solve for the goal of the MaxEnt classifier method for DF, i.e., the optimum global predictions, given solutions for $U_j$ (yielding either zeros or ones in this case):
    \[ U_j = \arg \max P(u_j / u_j^1, u_j^2, ..., u_j^N) \]

System Simulation and MOPs

- Measures of performance (MOPs) metrics are computed by enumeration of the prediction samples output the MaxEnt classifier driven by predictions sequences.
  - Given predictions sequences inputs, the probability of detection, $P_D$, is estimated as ratio of the sum of the number of received correct predictions (ones) of the classifier, divided by the sum of number of correct predictions (ones), based on the latest training data set.
  - Similarly, the probability of false alarm, $P_F$, estimated from ratio of the sum of number of correct predictions (ones) of the classifier divided by the of the sum of the number of false predictions (zeros) based on the latest training data set of the classifier.
  - In real-world applications above samples can be collected rapidly by taking a sequence of measurements at even MHz rates, and the storage requirements of one and zero sequences take up minimal memory space, at insignificant cost.
Example 2 E-MaxEnt: MaxEnt DDF Extension for Targets Classification/Recognition

An extension of MaxEnt DDF called “E-MaxEnt”, using distributed individual sensor classifiers for targets classification/ recognition, and fusing local classifier binary decisions at the parallel architecture fusion center.

**Purpose:** classify/recognize a previously seen “known” target in areas where previously seen targets most likely are, along with other targets. It should be noted that the approach is *not meant* to address the “needle-in-haystack” problem.

Unfortunately, the data sets to test the algorithm were not available, but front-end image processing and MaxEnt classifiers were implemented.
The Extended MaxEnt system (E-MaxEnt)

The elements of the system, depicted in E-MaxEnt Architecture are:

**Note:** The Binary Decisions (predictions), \( u_i \), are outputs of individual sensor classifiers, not the sensors

- (1) Multiple Sensors - a possible combination of both homogeneous and heterogeneous sensors, (e.g., EO, IR, HRR, ISAR, EO, etc.)
- (2) Sensor Dependent Detection and Pre-processing,
- (3) Sensor Dependent Feature Extraction,
- (4) MaxEnt Classifiers per sensor,
- (5) MaxEnt distributed fusion algorithm

Key differences between the MaxEnt and E-MaxEnt approaches are:

- E-MaxEnt doesn’t need to consider the rule that at least three sensors have to be active for DF. In E-MaxEnt each sensor is assumed to detect target signal for simultaneous target classification/recognition with synchronized sensors and overlapping FOV alike to MaxEnt, which is used for target detection only.
- Approach assumes association among multiple targets and multiple sensors have been solved, and the same targets are assigned in unison to the appropriate sensors.
The extended MaxEnt system (E-MaxEnt) Cont’d

Additional differences between the MaxEnt and E-MaxEnt approaches include:

- The decision fusion algorithm, is modified for use as the individual sensors classifier.
- Re-using the name of variables called binary decisions in Figure of the DF algorithm, to represent classifiers binary decisions (predictions), Eq. (6) is modified and replaced by Eq. (10), conditional probability below:

$$P(y_i | u_{ij}^1, u_{ij}^2, ... , u_{ij}^n) = \frac{\exp[\sum_{n=1}^N w_n f_n(u_{ij}^1, u_{ij}^2, ... , u_{ij}^n)]}{\sum_{x=0}^M \exp[\sum_{n=1}^N w_n f_n(u_{ij}^1, u_{ij}^2, ... , u_{ij}^n) = x]} \quad (10)$$

where, $f_n(u_{ij}^1, u_{ij}^2, ... , u_{ij}^n)$ is the $n^{th}$ feature function, $w_n$ is the weight parameter of the $n^{th}$ feature function to be estimated, and $x=m$ depends on the target class/type, and the historical (known/observed target) attribute measurement perspective of the target class/type.

- It provides the criteria to match feature functions constraints.
- For example, for radar cross-section measurement value at given aspect angle to match historical data of target class/type given the feature function for $m^{th}$ target class/type.
- $M$ target classes are defined a-priori.

The extended MaxEnt system (E-MaxEnt) Cont’d

The MaxEnt classifier algorithm continued:

- The hard feature function:

$$f_n(u_1, u_2, ... , u_n) = \begin{cases} 1, & \text{if } u_1 \prec u_2 \prec 4, u_2 \succ 3, u_4 \prec 5 \text{ where: } n \text{ is the } n^{th} \text{ feature function, and } x=k \text{ corresponds to the classifier declaration state 0 or 1.} \\
0, & \text{otherwise} \end{cases}$$

- The soft feature function (Example 2): it only equals one if certain constraints of the feature functions are satisfied, as shown below as an example:

$$f_n(u_1, u_2, u_3, u_4) = \begin{cases} 1, & \text{if } 3 < u_1 < 4, u_2 > 3, u_4 < 5 \text{ where: } n \text{ is the } n^{th} \text{ feature function, and } x=it, \text{ and } o.w.=0. \\
0, & \text{otherwise} \end{cases}$$

- A soft feature function for e.g., Hu’s seven moments (above) shows an initial ad-hoc-rule of constraints.
- It remains ad-hoc unless large data sets are available for classifier training, since one doesn’t know a-priory which moment feature is dominant.
The extended MaxEnt system (E-MaxEnt) Cont’d

- Large sets of data need to be collected from multiple sensor observations at different aspect angles and different environments of the previously seen (known) target whose presence is of interest.
- If collected measurements are not available, data needs to be generated using target models representing the previously seen (known) target models, at different aspect angles and different environments.
- Subsequently, the data becomes historical, and the learned rule will match the (known) target features function constraints.
- That is, one can also find the dominant moments, and reduce E-MaxEnt computational complexity and time.
- The two above approaches require testing and performance comparison.

MaxEnt Application to DF and (E-MaxEnt) Cont’d

- An alternative method to find dominant moments and further reduce E-MaxEnt computational complexity, one could use abductive logic/reasoning, which can be very complex.
- Therefore, instead human abductive reasoning, the use of a polynomial abductive Neural Network [36] requiring a small training set, should be able to rapidly yield the salient feature or feature set to use in the E-MaxEnt classifier, thereby reducing computational complexity.
- The efficacy of using the abductive NN feature MaxEnt preprocessor as a “feature set reducer”, needs to be evaluated and compared without its use of in the MaxEnt classifier.
- But having prior experience using abductive NNs one would expect overall reduced computational complexity. That is, the reduced feature set would significantly reduce E-MaxEnt computation time.
- In addition based on the proven excellent detection performance of MaxEnt, one would expect similar result in finding the presence, viz., recognizing a previously identified target of interest, among many similar targets in areas where previously seen targets most likely are, along with other targets.
Summary

Introduced:
• (1) The foundation, application, and efficacy of using MaxEnt for DDF
• (2) The framework and associated algorithms for E-MaxEnt
• Based on the proven excellent DDF detection performance of MaxEnt, one would expect similar result in finding the presence, viz., recognizing a previously known target of interest, among many similar targets in areas where previously seen targets most likely are, along with other targets.
• It is hoped that someone could provide the necessary data sets so the efficacy of the E-MaxEnt method could be demonstrated and compared with alternative approaches; and allow this endeavor.