REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY): 05-10-2012

2. REPORT TYPE: Conference Proceeding

3. DATES COVERED (From - To): -

4. TITLE AND SUBTITLE: Heterogeneous multi-metric learning for multi-sensor fusion

5a. CONTRACT NUMBER: W911NF-09-1-0383

5b. GRANT NUMBER:

5c. PROGRAM ELEMENT NUMBER: 611103

5d. PROJECT NUMBER:

5e. TASK NUMBER:

5f. WORK UNIT NUMBER:

6. AUTHORS

Author(s): Haichao Zhang , Nasrabadi, N.M., Huang, T.S., Yanning Zhang

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES

William Marsh Rice University
Office of Sponsored Research
William Marsh Rice University
Houston, TX 77005 -

8. PERFORMING ORGANIZATION REPORT NUMBER:

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S): ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S): 56177-CS-MUR.88

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

In this paper, we propose a multiple-metric learning algorithm to learn jointly a set of optimal homogenous/heterogeneous metrics in order to fuse the data collected from multiple sensors for classification. The learned metrics have the potential to perform better than the conventional Euclidean metric for classification. Moreover, in the case of heterogenous sensors, the learned multiple metrics can be quite different, which are adapted to each type of sensor. By learning the multiple metrics jointly within a single unified optimization

15. SUBJECT TERMS

metric learning , multi-sensor fusion

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UU	UU	UU	UU		Richard Baraniuk
					19b. TELEPHONE NUMBER: 713-348-5132

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

Report Title

Heterogeneous multi-metric learning for multi-sensor fusion

ABSTRACT

In this paper, we propose a multiple-metric learning algorithm to learn jointly a set of optimal homogenous/heterogeneous metrics in order to fuse the data collected from multiple sensors for classification. The learned metrics have the potential to perform better than the conventional Euclidean metric for classification. Moreover, in the case of heterogenous sensors, the learned multiple metrics can be quite different, which are adapted to each type of sensor. By learning the multiple metrics jointly within a single unified optimization framework, we can learn better metrics to fuse the multi-sensor data for joint classification.

Conference Name: Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on

Conference Date: July 05, 2011

Heterogeneous Multi-Metric Learning for Multi-Sensor Fusion

Haichao Zhang†‡
Beckman Institute
‡ University of Illinois at Urbana-Champaign, IL, 61801
Email: hczhang@ifp.uiuc.edu

Nasser M. Nasrabadi
U.S. Army Research Laboratory
2800 Powder Mill Road, Adelphi, MD, 20783

Thomas S. Huang
Beckman Institute
University of Illinois at Urbana-Champaign, IL, 61801
Email: huang@ifp.uiuc.edu

Yanning Zhang
School of Computer Science
† Northwestern Polytechnical University, Xi'an, China, 710129
Email: ynzhang@nwpu.edu.cn

***Abstract*—In this paper, we propose a multiple-metric learning algorithm to learn jointly a set of optimal homogenous/heterogeneous metrics in order to fuse the data collected from multiple sensors for classification. The learned metrics have the potential to perform better than the conventional Euclidean metric for classification. Moreover, in the case of heterogenous sensors, the learned multiple metrics can be quite different, which are adapted to each type of sensor. By learning the multiple metrics jointly within a single unified optimization framework, we can learn better metrics to fuse the multi-sensor data for joint classification.**

Keywords: metric learning, multi-sensor fusion.

I. Introduction

With advancement in sensor technology, numerous different kinds of sensors with diverse properties are being designed. A recent trend is to explore the abundant information from different sensors of homogenous or heterogeneous nature and fuse them for high-level decision such as classification. Multi-sensor fusion has applications ranging from daily life monitoring [2], [8], [18] to video surveillance [5], [10], and battle field monitoring and sensing [15], [18]. The use of multiple sensors has been shown to improve the robustness of the classification systems and enhance the reliability of the high-level decision making [2], [8], [10], [15], [18]. However, a direct challenge brought by using multiple sensors (heterogeneous or homogenous) is how to efficiently fuse the high-dimensional data deluge from these multiple sensors for high-level decision making (*e.g.*, classification). Li *et al.* [12] developed a general linear model unifying several different fusion architectures and also derived optimal fusion rules under several different scenarios. In [20], Varshney *et al.* developed a simultaneous linear dimension reduction and classifier learning algorithm for multi-sensor data fusion. In that algorithm, an alternating minimization scheme is adopted for achieving such a goal. To fuse the data from multiple sensors, the projected data for each sensor are concatenated and then used for training a classifier. Davenport *et al.* [5] proposed a joint manifold learning based method for data fusion by concatenating the data collected from multiple sensors using random projection as a universal dimensionality reduction scheme. In face of the increased complexity for parameter estimation in multi-sensor fusion, Lee *et al.* [11] developed a computationally efficient fusion algorithms based on Choleskey factorization.

Among many potential applications, we particularly focus on classification using multi-sensor fusion in this paper. At the core of many classification algorithms in pattern recognition is the notion of "distance". One of the most widely used methods is the k-nearest neighbor (KNN) method [4], which labels an input data sample to be the class with majority vote from its k-nearest neighbors. This method is non-parametric and is very effective and efficient for classification. Due to its effectiveness despite of its simplicity, it can be an effective candidate and can be easily extended to handle multiple sensors. Distance based method such as KNN relies on a proper definition of the distance metric to be most effective for the task at hand. This may be achieved based on the prior knowledge. However, in many cases where no such prior knowledge is available, a simple Euclidean metric is typically used for distance computation. Obviously, the Euclidean metric can not capture any of the regularities in the feature space of the data, thus it is sub-optimal for classification. To improve the performance of distance based classifiers, many algorithms have been developed to learn a proper metric for the application at hand in the past [6], [13], [21].

In the presence of multiple potentially heterogeneous sensors, the conventional metric learning method is not applicable due to its nature for single sensor. Although we can reduce the problem into a single metric learning problem by forming a long data vector constructed by concatenating data from all the sensors [20]. This method, however, poses great challenge to the learning algorithm due to the much higher dimensionality of the concatenated data vector. Another challenge brought by the multiple sensors is that how to fuse the information from all the sensors to improve the accuracy and reliability of the classification. In this paper, we develop a Homogenous/Heterogeneous Multi-Metric Learning (HMML) method to learn a metric set from multi-sensor training data, by exploiting the low-dimensional structures within the high-dimensional space. The proposed HMML method has computational advantage over the simple data concatenation method while it can exploit the correlations among the multiple sensors during metric learning procedure. Based on the learned metric set, an energy based classification method is adopted which uses the learned sensor-specific metrics and naturally fuses all the information from all the sensors for a single, joint classification decision.

The rest of this paper is organized as follows. In Section II, we review briefly some related works on metric learning. In Section III, we introduce the Heterogeneous Multi-Metric Learning method and present an efficient algorithm for it. Extensive experiments using real multi-sensor datasets are carried out in Section V to verify the effectiveness of the proposed method. We make some discussions and conclude the paper in Section VI.

II. Metric Learning: PCA, LDA and LMNN

We first review briefly some related methods for learning an optimal metric for a single sensor under different criteria. We use $\mathbf{x}$ to denote a data sample. A family of metrics can be induced by a linear transformation (feature extraction) operator $\mathbf{P}$ as $\tilde{\mathbf{x}} = \mathbf{P}\mathbf{x}$ followed by using Euclidean metric in the transformed space. Specifically, the squared distance in the space after linear transformation using $\mathbf{P}$ is calculated as:

$$\begin{aligned} d(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) &= d_{\mathbf{P}}(\mathbf{x}_i, \mathbf{x}_j) \\ &= \|\mathbf{P}\mathbf{x}_i - \mathbf{P}\mathbf{x}_j\|_2^2 = \|\mathbf{P}(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 \\ &= (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{P}^\top \mathbf{P}(\mathbf{x}_i - \mathbf{x}_j) \\ &= (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j), \end{aligned} \quad (1)$$

where $\mathbf{M} = \mathbf{P}^\top\mathbf{P}$. Therefore, a linear transformation $\mathbf{P}$ can introduce a Mahalanobis metric $\mathbf{M} = \mathbf{P}^\top\mathbf{P}$ in the original space, thus we also denote $d_{\mathbf{P}}(\mathbf{x}_i, \mathbf{x}_j)$ as $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$ according to the specific parametrization we adopt. In this paper, we use linear projection model due to its simplicity as well as its effectiveness. Under this model, the problem of metric learning for $\mathbf{M}$ is equivalent to learning the linear projection operator $\mathbf{P}$.

The following notations are used in this work. We use $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ to denote the set of training samples, where $\mathbf{x}_i \in \mathbf{R}^d$ is the i-th data sample while $y_i \in \{1, 2, \cdots, C\}$ is its corresponding label. In presence of multiple sensors, we use $\left\{(\{\mathbf{x}_i^s\}_{s=1}^S, y_i)\right\}_{i=1}^N$ to denote the set of training samples, where each training sample is actually a set $\{\mathbf{x}_i^s\}_{s=1}^S$ consisting of all the i-th data samples from S different sensors.

A. Principal Component Analysis

One of the most well-known projection method is the Principal Component Analysis (PCA) method [9] which seeks a projection matrix $\mathbf{P}$ by maximizing the variance after projection (thus retaining the maximum energy), which can be achieved via:

$$\begin{aligned} \mathbf{P} = \arg\max_{\mathbf{P}} & \ \mathrm{Tr}(\mathbf{P}^\top\mathbf{C}\mathbf{P}) \\ \text{s.t.} & \quad \mathbf{P}\mathbf{P}^\top = \mathbf{I}, \end{aligned} \quad (2)$$

where $\mathbf{C}$ is the covariance matrix of the data. (2) has closed-form solution which states that the rows of $\mathbf{P}$ are constructed as the leading eigenvectors of $\mathbf{C}$. PCA captures the low-dimensional property of the data by seeking the projection directions keeping most of the variance/energy of all the data samples from all classes. Therefore, the induced metric $\mathbf{M}$ is a low rank matrix which eliminates the components with low energies. By learning the projection/metric in this way, the learned projection/metric is good for reconstruction of the data, but it is not necessarily effective for classification.

B. Linear Discriminant Analysis

To introduce discriminative power into the projection, the linear discriminate analysis (LDA) method [14] is used to obtain discriminative projections by maximizing the between-class scattering while minimizing the within-class variance. This can be achieved via:

$$\begin{aligned} \mathbf{P} = \arg\max_{\mathbf{P}} & \ \mathrm{Tr}\left(\frac{\mathbf{P}^\top\mathbf{C}_b\mathbf{P}}{\mathbf{P}^\top\mathbf{C}_w\mathbf{P}}\right) \\ \text{s.t.} & \quad \mathbf{P}\mathbf{P}^\top = \mathbf{I}, \end{aligned} \quad (3)$$

where $\mathbf{C}_b$ and $\mathbf{C}_w$ are between-class and within-class covariance matrix respectively. The projection matrix $\mathbf{P}$ can be obtained as the leading eigenvectors of $\mathbf{C}_w^{-1}\mathbf{C}_b$ (assuming $\mathbf{C}_w$ is invertible). By incorporating the label information from each sample into the optimization, the learned metric $\mathbf{M}$ is better suited for discrimination. Both PCA and LDA can be viewed under a unified framework called Graph Embedding [23], which can be applied with eigen-analysis with different configurations of intrinsic graphs and penalty graphs to generate different projection matrix $\mathbf{P}$, thus inducing metrics with different properties.

C. Large-Margin Nearest Neighbor Metric Learning

Apart from the eigen-analysis based methods, another line of research for metric learning is via convex optimization, typically formulated as a semi-definite programming (SDP) problem [6], [13], [21]. A representative example is the Large Margin Nearest Neighbor (LMNN) method [21] which will be briefly reviewed in the sequel. LMNN method tries to learn an optimal metric specifically for KNN classifier. The basic idea is to learn a metric under which the k nearest neighbors for a training sample are samples belonging to the same class as the test sample. LMNN method relies on two intuitions to learn such a metric: (1) each training sample should have the same label as its k nearest neighbors; (2) training samples with different labels should be far from each other. To formulate the above intuitions formally, Weinberger *et al.* [21] introduced the following two energy terms:

$$E_{\mathrm{pull}}(\mathbf{P}) = \sum_{i,j \rightsquigarrow i} \|\mathbf{P}(\mathbf{x}_i - \mathbf{x}_j)\|^2, \quad (4)$$

$$\begin{aligned} E_{\mathrm{push}}(\mathbf{P}) = \sum_{i,j \rightsquigarrow i} \sum_{l} (1 - y_{il}) \Big[& 1 + \|\mathbf{P}(\mathbf{x}_i - \mathbf{x}_j)\|^2 \\ & - \|\mathbf{P}(\mathbf{x}_i - \mathbf{x}_l)\|^2 \Big]_+, \end{aligned} \quad (5)$$

where i indexing the training samples and $j \rightsquigarrow i$ denotes the set of 'target' neighbors of $\mathbf{x}_i$, *i.e.*, the k nearest samples with the same label as $\mathbf{x}_i$. $y_{il} \in \{0, 1\}$ is a binary number indicating whether $\mathbf{x}_i$ and $\mathbf{x}_l$ are of the same class. $[\cdot]_+ = \max(\cdot, 0)$ is a hinge loss. The samples contributing to the energy $E_{\text{push}}(\mathbf{P})$ are termed as 'impostors', which are in fact those samples within the radius of target samples but belong to classes different from the target class.

$E_{\text{pull}}(\mathbf{P})$ is the energy function giving large energy to the large distances of the KNN samples belonging to the same class (target samples) while $E_{\text{push}}(\mathbf{P})$ is the energy function quantifying the energy between samples from different classes (impostors), which gives large energy to the small distance KNN samples from a different class. To learn a metric under which the target samples are near to each other while the impostors are far from each other, the following total energy was proposed by Weinberger *et al.* in [21]:

$$E(\mathbf{P}) = (1 - \lambda)E_{\text{pull}}(\mathbf{P}) + \lambda E_{\text{push}}(\mathbf{P}), \tag{6}$$

where $0 \leq \lambda \leq 1$ is the parameter balancing the two terms. In practice, we set $\lambda = 0.5$ which gives good results. This loss function is not convex, therefore, in [21] they reformulated the original problem into a SDP problem as follows:

$$\begin{aligned} \mathbf{M} = \arg\min_{\mathbf{M}} (1 - \lambda) &\sum_{i,j \rightsquigarrow i} (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) \\ + \lambda &\sum_{i,j \rightsquigarrow i} \sum_{l} (1 - y_{il}) \epsilon_{ijl} \\ \text{s.t. } & (\mathbf{x}_i - \mathbf{x}_l)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_l) - (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) \geq 1 - \epsilon_{ijl} \\ & \epsilon_{ijl} \geq 0, \quad \mathbf{M} \succeq 0, \end{aligned}$$

where $\mathbf{M} = \mathbf{P}\mathbf{P}^\top$. The set of target samples in LMNN can be initialized with Euclidean metric and be fixed during the learning process [21]. Extension of LMNN to learning multiple local metrics has been made in [22] by learning a specific metric within a local cluster of features. Very recently, LMNN has been generalized into multi-task setting, where the multiple tasks for metric learning are coupled by a 'common' metric shared by all the tasks and an additive 'innovative' metric that is specific for each task [17].

III. Heterogeneous Multi-Metric Learning based Multi-Sensor Fusion for Classification

In this section, we will develop the Heterogeneous Multi-Metric Learning (HMML) method for multi-sensor fusion based classification. Similar to the single sensor metric learning case, we develop our HMML method for multi-sensor data based on two similar intuitions as follows: (1) each training sample should have the same label as its k nearest neighbor in the full feature space; (2) training samples with different labels should be far from each other in the full feature space. Given N training samples from S potentially heterogeneous sensors $\big\{(\{\mathbf{x}_i^s\}_{s=1}^S, y_i)\big\}_{i=1}^N$, we aim to learn a metric (projection) set $\{\mathbf{P}^s\}_{s=1}^S$ for the multiple sensors, where $\mathbf{P}^s$ is the projection matrix for the s-th sensor. Learning the metric in such a heterogeneous way jointly, we can adapt the metric to each sensor more suitably and improve the robustness of final joint classification. Following the same spirit as LMNN, we propose the following 'pull' and 'push' energy terms for multiple sensors:

$$E_{\text{pull}}(\{\mathbf{P}^s\}_{s=1}^S) = \sum_{i,j \rightsquigarrow i} \sum_{s=1}^{S} \left\| \mathbf{P}^s (\mathbf{x}_i^s - \mathbf{x}_j^s) \right\|^2. \tag{7}$$

$$\begin{aligned} E_{\text{push}}(\{\mathbf{P}^s\}_{s=1}^S) = \sum_{i,j \rightsquigarrow i} \sum_{l} (1 - y_{il}) \Big[1 + \sum_{s=1}^{S} \left\| \mathbf{P}^s (\mathbf{x}_i^s - \mathbf{x}_j^s) \right\|^2 \\ - \sum_{s=1}^{S} \| \mathbf{P}^s (\mathbf{x}_i^s - \mathbf{x}_l^s) \|^2 \Big]_+ . \end{aligned} \tag{8}$$

The hinge loss $[\cdot]_+$ used in (8) couples the multiple metrics and enables them to be learned jointly from the training data, thus fusing the information from all the sensors by learning appropriate metrics adapted to each sensor. Using these energy terms, the total energy is defined as:

$$E\big(\{\mathbf{P}^s\}_{s=1}^S\big) = (1 - \lambda) E_{\text{pull}}\big(\{\mathbf{P}^s\}_{s=1}^S\big) + \lambda E_{\text{push}}\big(\{\mathbf{P}^s\}_{s=1}^S\big). \tag{9}$$

Again, (9) is not convex. To solve it effectively, we reformulate it into a SDP problem following LMNN:

$$\begin{aligned} \{\mathbf{M}^s\} = \arg\min_{\{\mathbf{M}^s\}} (1 - \lambda) &\sum_{i,j \rightsquigarrow i} \sum_{s=1}^{S} (\mathbf{x}_i^s - \mathbf{x}_j^s)^\top \mathbf{M}^s (\mathbf{x}_i^s - \mathbf{x}_j^s) \\ + \lambda &\sum_{i,j \rightsquigarrow i} \sum_{l} (1 - y_{il}) \epsilon_{ijl} \\ \text{s.t. } &\sum_{s=1}^{S} \big\{ (\mathbf{x}_i^s - \mathbf{x}_l^s)^\top \mathbf{M}^s (\mathbf{x}_i^s - \mathbf{x}_l^s) \\ &\quad - (\mathbf{x}_i^s - \mathbf{x}_j^s)^\top \mathbf{M}^s (\mathbf{x}_i^s - \mathbf{x}_j^s) \big\} \geq 1 - \epsilon_{ijl} \\ & \epsilon_{ijl} \geq 0, \quad \mathbf{M}^s \succeq 0, \end{aligned} \tag{10}$$

where $\mathbf{M}^s = \mathbf{P}^s \mathbf{P}^{s\top}$. By converting the original problem into a SDP problem, it can be easily solved via standard SDP solvers. The detailed algorithm for solving this problem is presented in the next section.

After the metric set $\{\mathbf{M}^s\}_{s=1}^S$ is learnt, we can proceed to perform classification by fusing the information from all the sensors. Given a multi-sensor test sample $\mathbf{x}_t = \{\mathbf{x}_t^s\}_{s=1}^S$, we can classify it using a KNN classifier with the learned metrics. Alternatively, the following energy based classification method can be used for better classification performance [21]. Denoting the distance between the multi-sensor test sample $\mathbf{x}_t$ and a multi-sensor training sample $\mathbf{x}_i = \{\mathbf{x}_i^s\}_{s=1}^S$ as

$$D_{\mathbf{M}}(\mathbf{x}_t, \mathbf{x}_i) = \sum_{s=1}^{S} d_{\mathbf{M}^s}(\mathbf{x}_t^s, \mathbf{x}_i^s), \tag{11}$$

the energy based classification can be achieved via [21]:

$$\begin{aligned}\hat{y}_t = \arg\min_{y_t}(1-\lambda)\sum_{j\rightsquigarrow t} D_{\mathbf{M}}(\mathbf{x}_t,\mathbf{x}_j) \\ +\lambda\sum_{j\rightsquigarrow t,l}(1-y_{tl})\Big[1+D_{\mathbf{M}}(\mathbf{x}_t,\mathbf{x}_j)-D_{\mathbf{M}}(\mathbf{x}_t,\mathbf{x}_l)\Big]_+ \\ +\lambda\sum_{i,j\rightsquigarrow i}(1-y_{it})\Big[1+D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j)-D_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_t)\Big]_+.\end{aligned} \tag{12}$$

The first term in (12) represents the accumulated energy for the k target neighbors of $\mathbf{x}_t$; The second term accumulates the hinge loss over all the imposters for $\mathbf{x}_t$; the third term represents the accumulated energy for different labeled samples whose neighbor perimeters are invaded by $\mathbf{x}_t$, *i.e.*, taking $\mathbf{x}_t$ as their imposter.

IV. Efficient Heterogeneous Multi-Metric Learning Algorithm

After we get the SDP formulation (10), a general purpose SDP solver can be used to solve the multi-metric learning problem. However, as the general purpose solvers do not take the special structures of the problem into consideration, they do not scale well in the number of constraints. Following [22], we also exploit the fact that most of the constraints are not active, *i.e.*, most of slack variables $\{\epsilon_{ijl}\}$ never have positive values. Therefore, by using only the sparse active constraints, a great speedup can be achieved. An efficient algorithm for HMML is developed in this section. The main algorithm includes two key steps: (1) gradient descent of the metrics and (2) projection onto the SDP cone. We address each of these aspects in the following.

1) Gradient Computation: By using the notation $\mathbf{C}_{ij}^s = (\mathbf{x}_i^s - \mathbf{x}_j^s)(\mathbf{x}_i^s - \mathbf{x}_j^s)^\top$. At the t-th iteration, we have $D_{\mathbf{M}}^t(\mathbf{x}_i,\mathbf{x}_j) = \sum_{s=1}^S \mathrm{tr}(\mathbf{M}_t^s\mathbf{C}_{ij}^s)$. Therefore, we can reformulate the energy function (9) as:

$$E(\{\mathbf{M}_t^s\}_{s=1}^S) = (1-\lambda)\sum_{i,j\rightsquigarrow i}\sum_s \mathrm{tr}(\mathbf{M}_t^s\mathbf{C}_{ij}^s) \tag{13}$$

$$+\lambda\sum_{i,j\rightsquigarrow i}\sum_l(1-y_{il})\Big[1+\sum_s\big(\mathrm{tr}(\mathbf{M}_t^s\mathbf{C}_{ij}^s)-\mathrm{tr}(\mathbf{M}_t^s\mathbf{C}_{il}^s)\big)\Big]_+.$$

We define a set of triples $\mathcal{N}_t$ as the set of indices $(i,j,l)\in\mathcal{N}_t$ if and only if (i,j,l) triggers the hinge loss in (13), which is also referred to as *active set* in the following. The gradient of (13) with respect to $\mathbf{M}_t^s$ is:

$$\begin{aligned}\mathbf{G}_t^s &= \frac{\partial E(\{\mathbf{M}_t^s\}_{s=1}^S)}{\partial \mathbf{M}_t^s} \\ &= (1-\lambda)\sum_{i,j\rightsquigarrow i}\mathbf{C}_{ij}^s \quad +\lambda\sum_{(i,j,l)\in\mathcal{N}_t}(\mathbf{C}_{ij}^s-\mathbf{C}_{il}^s).\end{aligned} \tag{14}$$

Note that the updating of $\mathbf{G}^s$ requires the computation of the outer product in $\mathbf{C}_{ij}^s$. This updating step may be computationally expensive. Thus we use an active updating scheme following [22]:

$$\begin{aligned}\mathbf{G}_{t+1}^s = \mathbf{G}_t^s - \lambda\sum_{(i,j,l)\in\mathcal{N}_t-\mathcal{N}_{t+1}}(\mathbf{C}_{ij}^s-\mathbf{C}_{il}^s) \\ +\lambda\sum_{(i,j,l)\in\mathcal{N}_{t+1}-\mathcal{N}_t}(\mathbf{C}_{ij}^s-\mathbf{C}_{il}^s).\end{aligned} \tag{15}$$

This means that to get an updated estimation for the next estimation of the gradient corresponding to sensor s, we subtract the contribution of the inactive samples ($\mathcal{N}_t-\mathcal{N}_{t+1}$, *i.e.*, the samples contained in $\mathcal{N}_t$ but not in $\mathcal{N}_{t+1}$) from the previous gradient estimation and add the contribution of the newly activated samples ($\mathcal{N}_{t+1}-\mathcal{N}_t$, *i.e.*, the samples contained in $\mathcal{N}_{t+1}$ but not in $\mathcal{N}_t$) from sensor s. In the presence of multiple sensors, the active sample set $\mathcal{N}_t$ has to be updated based on the data from all the sensors, thus fusing them effectively and ensuring a more effective updating step for all the metrics. This step exploits the correlations among the potentially heterogeneous data from the multiple sensors and can improve the performance of the algorithm, both in terms of classification accuracy and robustness, as verified by the experimental results in the next section.

2) Projection: The minimization of (9) or (10) must enforce that the metric $\mathbf{M}_t^s$ should be positive semi-definite. This is approached by projecting the current estimation onto the cone of all positive semidefinite matrices $\mathcal{S}_+$. For the current estimation of the metric $\mathbf{M}_t^s$ for sensor s, we perform eigen-decomposition:

$$\mathbf{M}_t^s = \mathbf{V}\Delta\mathbf{V}^\top, \tag{16}$$

where $\mathbf{V}$ consists of the eigenvectors of $\mathbf{M}_t^s$ and Δ is a diagonal matrix with corresponding eigen-values. The projection of $\mathbf{M}_t^s$ onto the SDP cone is implemented as:

$$\mathcal{P}_{\mathcal{S}}(\mathbf{M}_t^s) = \mathbf{V}\Delta_+\mathbf{V}^\top, \tag{17}$$

where $\Delta_+ = \max(\Delta, 0)$.

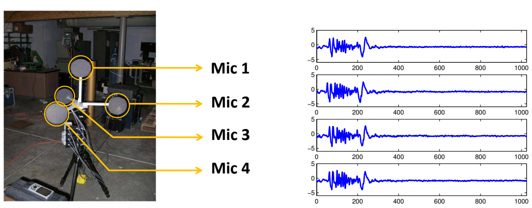
Using the derived gradient updating equation (15) and the SDP projection (17), the multi-metric learning procedure can be implemented by taking a gradient descent step at each iteration and then projecting back onto the SDP cone for each sensor specific metric based on the active set updated using all the sensors, thus fusing the information from multiple sensors. The overall learning procedure is summarized in Algorithm 1.

V. Multi-Sensor Acoustic Signal Fusion for Event Classification

In this section, we carry out experiments on a number of real acoustic datasets and compare the results with several conventional classification methods to verify the effectiveness of the proposed method. Specifically, we first show an illustrative example to demonstrate some properties of the proposed method. We then examine the advantage of learning multiple metrics jointly as proposed. The merits of using a joint multi-metric in multi-sensor classification is then examined. Furthermore, we test the proposed method on a 2-class classification problem, and then on a 4-class classification

Algorithm 1: Heterogeneous Multi-Metric Learning (HMML).

Input: multi-sensor data training set $\{(\{\mathbf{x}_i^s\}_{s=1}^S, y_i)\}_{i=1}^N$, number of nearest neighbor k, gradient step length α, weight λ
Output: multi-sensor metric set $\{\mathbf{M}^s\}_{s=1}^S$
Initialize: $\{\mathbf{M}^s\}_{s=1}^S = \mathbf{I}$, $\mathbf{G}_0^s \leftarrow (1-\lambda)\sum_{i,j\rightsquigarrow i} \mathbf{C}_{ij}^s$, $t \leftarrow 0$, $\mathcal{N}_t = \{\}$;
while *convergence condition false* **do**
  Update the active set $\mathcal{N}_{t+1}$ by collecting the triplets (i,j,l) with $j \rightsquigarrow i$ that incur the hinge loss in (13);
  for $s = 1, 2, \cdots, S$ **do**
    % compute the gradient to the metric for sensor s
    $\mathbf{G}_{t+1}^s \leftarrow \mathbf{G}_t^s - \lambda \sum_{(i,j,l)\in\mathcal{N}_t - \mathcal{N}_{t+1}} (\mathbf{C}_{ij}^s - \mathbf{C}_{il}^s) + \lambda \sum_{(i,j,l)\in\mathcal{N}_{t+1} - \mathcal{N}_t} (\mathbf{C}_{ij}^s - \mathbf{C}_{il}^s)$;
    % take gradient step and project onto SDP cone for the metric of the s-th sensor
    $\mathbf{M}_{t+1}^s \leftarrow \mathcal{P}(\mathbf{M}_t^s - \alpha \mathbf{G}_{t+1}^s)$;
  end
  $t \leftarrow t+1$
end



(a) (b)

Figure 1. Illustration of multiple sensors and the multi-sensor data. (a) UTAMS acoustic sensor array. Each array has 4 acoustic sensors, collecting multiple acoustic signals of the same physical event simultaneously. (b) acoustic signals from the 4 acoustic sensors for a Rocket Launch event collected by a UTAMS.

problem. To examine the effects caused by different number of training samples, we also carry out experiments under different training ratios. Finally, to evaluate the effects of physical sites on classification, we carry out experiments with multi-sensor data collected from different sites for training and testing.

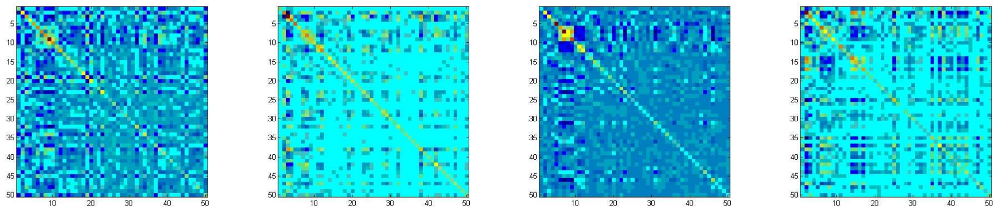
Data Description: The multi-sensor transient acoustic data is collected for launch and impact of different weapons (mortar and rocket) using the Unattended Transient Acoustic MASINT System (UTAMS) developed by the U.S. Army Research Laboratory as shown in Figure 1. For each event, a UTAMS measures the signal from a launch/impact event using 4 acoustic sensors simultaneously, where the sampling rate is 1001.6 Hz. Totally, we have 4 datasets: CRAM04, CRAM05, CRAM06 which were collected on different years, and another dataset called Foreign which contains acoustic signals of foreign weapons [16]. Among these 4 datasets, CRAM05 and Foreign datasets consist of 4 subsets collected by UTAMS sensors deployed at 4 different physical sites.

Segmentation: The event can occur at arbitrary location of the raw acoustic signal. We first segment the raw signal with spectral maximum detection [7] and then extract the appropriate features from those segmented signals. In our experiments, we take a segment with 1024 sampling points.
Feature Extraction: We use Cepstral features [3] for classification, which have been proved to be effective in speech and acoustic signal classification. We discard the first Cepstral coefficient and keep the following 50 Cepstral coefficients.

To evaluate the effectiveness of the proposed method, we compare the results with different classical algorithms including sparse linear multinomial Logistic Regression [1], [19] and Linear Support Vector Machine (SVM) [1], which runs in two modes in our experiments: (1) treating each sensor signal separately (SVM); (2) concatenating all the signals from different sensors (CSVM). One-vs-all scheme is used for SVM in the case of multi-class classification. To show the improvement by learning the metric, we also compare the results with the classification results with model (12) using Euclidean metric, which is denoted as KNN in the sequel.

A. Heterogeneous Multi-Metric Learning: An Example

We first illustrate some features of the proposed HMML algorithm on a 2-class classification problem with 4 acoustic sensors using the CRAM04 dataset. The 2-class classification problem is defined as discriminating between different event types (launch/impact) of a specific weapon (mortar). We first examine the effect of the number of nearest neighbor k on the classification accuracy. We carry out experiments under different number of nearest neighbors: $k \in \{3, 5, 7, 9\}$ with training ratio $r = 0.5$ (the ratio of the number training samples with respect to that of the whole dataset) and summarize the results in Table I. As can be seen from Table I, the proposed HMML method outperforms the other methods under different number of nearest neighbors. Note that using KNN method directly gives results worse than those of SVM. However, after learning multiple metrics using the proposed method, a large improvement in the classification accuracy over KNN is gained. The proposed HMML method performs better than SVM as well as CSVM under different number of nearest neighbors. Note that the proposed method is also robust to the number of nearest neighbor k, as shown in Table I. We set $k = 3$ in the following experiments unless otherwise specified. The 4 metrics (induced by the projection operation) learned for each acoustic sensor via the proposed method are shown in Figure 2. As can be seen from Figure 2, the 4 learned metrics are with some similar diagonal patterns, due to the joint learning process. However, as can be noticed from Figure 2, these 4 learned metrics as adapted to each sensor are not exactly the same in nature, although they are all learned for acoustic sensors, which implies that the 4 acoustic sensors may have different operating conditions and contribute differently to classification. The metrics learned for sensor 1 and sensor 3 (sensor 2 and sensor 4) are similar to each other, indicating potentially similar operating conditions for those sensors. Some of the learned metrics have different property



(a) Sensor 1 (b) Sensor 2 (c) Sensor 3 (d) Sensor 4

Figure 2. Illustration of the learned metrics $\{\mathbf{M}^s\}_{s=1}^4$ for the 4 acoustic sensors for 2 class Mortar problem (4 sensors).

with each other, indicating that even though the sensors are all acoustic sensors, thus homogenous, the learned metrics which are adapted to each sensor can be heterogeneous. By learning such a heterogeneous metric set combining the cues from all the sensors, we can learn multiple heterogeneous metrics adapted to each sensor with improved classification performance via the joint learning process. Therefore, the proposed HMML algorithm is more robust and flexible to the sensor fusion classification task.

Table I
CLASSIFICATION ACCURACY FOR WITH INCREASING NUMBER OF NEAREST NEIGHBORS k (2-CLASS MORTAR PROBLEM USING CRAM04 DATASET, $S = 4, r = 0.5$).

k	3	5	7	9
Logistic	0.7778	0.7778	0.7778	0.7778
SVM	0.8073	0.8073	0.8073	0.8073
CSVM	0.8173	0.8173	0.8173	0.8173
KNN	0.6808	0.6840	0.6821	0.6821
HMML	**0.8673**	**0.8644**	**0.8644**	**0.8490**

B. *The Merits of Joint Learning of Multiple Metrics*

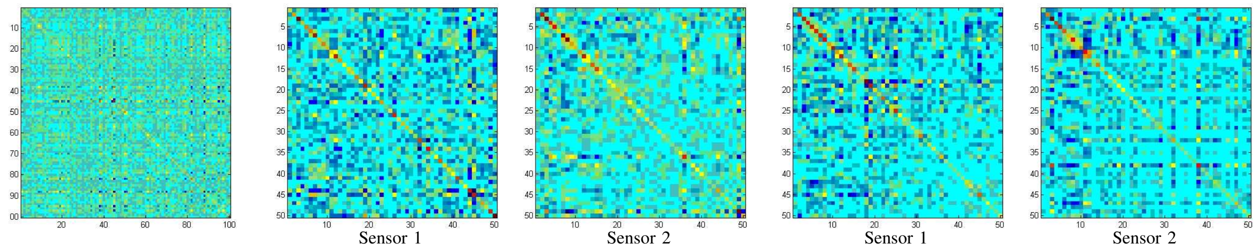
In this subsection, we conduct several experiments to verify the advantages of the proposed joint approach for learning multiple metrics. For comparison, we also learn the metrics with (i) Separate Metric Learning (SML): learning a metric for each sensor separately; (ii) Concatenated Metric Learning (CML): learning the metric using data formed by concatenating the data from multiple sensors. The classification results under training ratio $r = 0.5$ for two class mortar problem with different number of sensors ($S \in \{1, 2, 3, 4\}$) using CRAM04 dataset are shown in Table II. As can be seen from Table II, all the three metric learning methods (SML, CML and HMML) can substantially improve the classification performance over the method without metric learning (KNN). However, by learning the multiple metrics jointly using the proposed HMML method, we can achieve better classification accuracy than the method of learning metrics separately for each sensor (SML) and the method of concatenating the data (CML). The learned metrics using different methods are shown in Figure 3. As can be seen from this figure, the metrics learned using different methods are very different. Although CML can improve the classification accuracy over KNN by a notable amount, by concatenating the data and learn the corresponding metric in that high-dimensional space, the dimensionality of the learning task has been increased by a large number, which poses great challenge to the learning algorithm. Moreover, the computational demand is also increased to learn a full and dense metric matrix in the enlarged data space. By learning one metric for each sensor separately using SML, the learning problem suffers less from the curse-of-dimensionality and is also less demanding in computation, while achieving similar performance with CML, as shown in Table II. The problem with SML is that it totally overlooks the correlations among the data from multiple sensors, therefore it can not exploit these correlations during metric learning to improve its performance, thus is not the most effective scheme for sensor fusion (see Figure 3 (b)). Using the proposed HMML method to learn the metric jointly, we can enjoy computational efficiency while exploiting the correlations among the data from multiple sensors. Thus obtaining a metric set that is more discriminative in classification and improving the classification accuracy over SML and CML by a large margin, as shown in Table II. The metrics learned using proposed HMML method are shown in Figure 3 (c).

Table II
COMPARISON OF JOINT AND SEPARATE METRIC LEARNING (2-CLASS MORTAR PROBLEM USING CRAM04 DATASET, $k = 3, r = 0.5$).

Number of Sensors S	1	2	3	4
KNN	0.6820	0.6817	0.6808	0.6837
SML	**0.8170**	0.7958	0.8131	0.8025
CML	**0.8170**	0.7987	0.8039	0.8183
HMML	**0.8170**	**0.8600**	**0.8644**	**0.8673**

C. *The Merits of Using Multiple Sensors*

In this subsection, we examine the effects of fusing data from multiple sensors on classification compared with using only data from a single sensor. We again use the two class mortar problem on the CRAM04 dataset as an example. We vary the number of sensors within the range $S \in \{1, 2, 3, 4\}$ and carry out classification experiments using different algorithms with training ratio $r = 0.5$. For Logistic regression and SVM, they are performed on each sensor separately and the average performances are reported. For CSVM, concatenated data from all the sensors are used for classification. The experimental results are presented in Table III and also graphically depicted in Figure 4. As can be seen from these results, the classification accuracy increase as the number of sensor increase in general. The proposed HMML method is comparable to other methods in the case of using single sensor ($S = 1$)



(a) Concatenate (b) Separate (c) Joint

Figure 3. Comparison of the metrics learned via different approaches (2-class Mortar problem using CRAM04 dataset with 2 sensors): (a) concatenating the data (b) separately for each sensor (c) jointly for all the sensors.

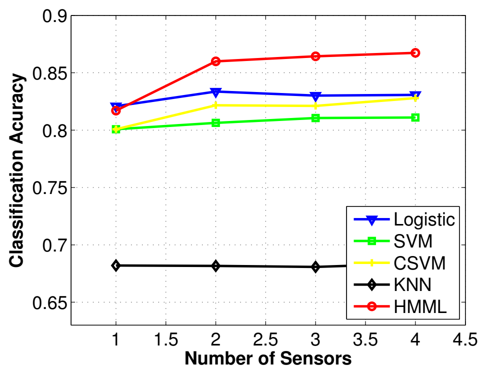


Figure 4. Accuracy curves for different algorithms with increasing number of sensors S (2-class Mortar problem using CRAM04 dataset).

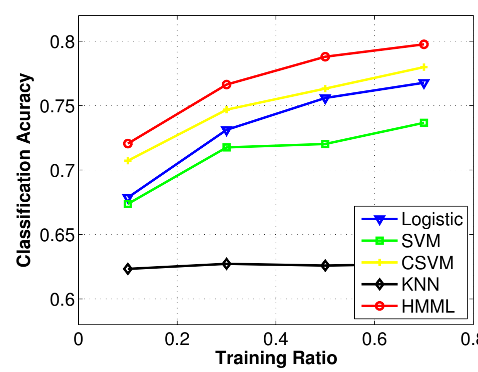


Figure 5. Accuracy curves for different algorithms with increasing training ratio r (4-class problem using CRAM04 dataset).

for classification. In the case of multiple sensors ($S \geq 2$), our HMML method outperforms all the other methods by a notable margin. Specifically, with the learned metric, HMML improves the classification accuracy signaficantly over KNN, which uses Euclidean metric for classification.

Table III
CLASSIFICATION ACCURACY USING DATA FROM DIFFERENT NUMBER OF SENSORS (2-CLASS MORTAR PROBLEM USING CRAM04 DATASET, $k = 3, r = 0.5$).

Number of Sensors S	1	2	3	4
Logistic	0.8209	0.8336	0.8301	0.8307
SVM	0.8008	0.8063	0.8106	0.8111
CSVM	0.8008	0.8217	0.8211	0.8279
KNN	0.6820	0.6817	0.6808	0.6837
HMML	**0.8170**	**0.8600**	**0.8644**	**0.8673**

D. Two Class Event Classification

In this experiment, we focus on the classification problem between launch and impact for a single kind of weapon (mortar) using all the 4 datasets. We randomly split each dataset into two subsets for training and testing, with training ratio $r = 0.5$. We run the experiment 5 times and summarize the average performance in Table IV for different datasets. As can be seen from comparison, HMML performs better than Logistic regression or linear SVM and improves the performance over KNN significantly, which clearly demonstrates the effectiveness of the proposed multi-sensor metric

Table IV
CLASSIFICATION ACCURACY FOR 2-CLASS MORTAR PROBLEM ($S = 4, k = 3, r = 0.5$).

Method	04	05	06	Foreign	Average
Logistic	0.7778	0.8069	0.7183	0.6857	0.7472
SVM	0.8073	0.7991	0.7917	0.7693	0.7919
CSVM	0.8173	0.8448	0.7938	0.8000	0.8140
KNN	0.6808	0.8241	0.6949	0.7800	0.7450
HMML	**0.8673**	**0.8621**	**0.8525**	**0.8240**	**0.8515**

learning method. Moreover, it is noticed that for KNN, its performance varies a lot from one dataset to another dataset, while the proposed HMML method performs equally on different datasets, which implies its robustness and potential applicability to real-world problems.

Table V
CLASSIFICATION ACCURACY FOR 4-CLASS PROBLEM ($S = 4, k = 3, r = 0.5$).

Method	04	05	06	Foreign	Average
Logistic	0.7440	0.7234	0.6882	0.7367	0.7231
SVM	0.7410	0.7227	0.6860	0.7474	0.7243
CSVM	0.7487	**0.7375**	0.6945	0.7169	0.7244
KNN	0.6204	0.7188	0.6236	0.7456	0.6771
HMML	**0.8014**	0.7313	**0.7284**	**0.7928**	**0.7635**

E. Four Class Event Classification

To further verify the effectiveness of the proposed method, we test our algorithm on a 4-class classification problem, where we want to make decision on whether the event is

Table VI
CLASSIFICATION ACCURACY FOR 4-CLASS CLASSIFICATION WITH TRAINING AND TESTING ON DATA MEASURED AT DIFFERENT PHYSICAL SITES ($S = 4, k = 3$).

Method	CRAM05				Foreign				Average
	Site 1	Site 2	Site 3	Site 4	Site 1	Site 2	Site 3	Site 4	
Logistic	0.4605	0.5577	0.6750	0.6172	0.6797	0.6829	0.7314	0.7109	0.6394
SVM	0.4408	0.5577	0.6583	0.6328	0.6901	0.7134	**0.8005**	0.6652	0.6449
CSVM	0.5526	0.6538	0.6667	0.4688	0.7292	0.7073	0.7766	0.7043	0.6574
KNN	0.4737	0.8462	0.5333	0.6250	0.5104	0.4146	0.5000	0.6348	0.5673
HMML	**0.5789**	**0.8846**	**0.8000**	**0.7500**	**0.7917**	**0.7805**	0.7766	**0.7391**	**0.7627**

launch or impact and whether the weapon is mortar or rocket, which is much more challenging. We generate training and testing datasets by random sampling each dataset with training ratio $r = 0.5$. We repeat the experiment 5 times and report the average performance for each dataset as well as the overall average classification accuracy in Table V. We can see again that the proposed HMML method performs better than all the other methods and outperforms KNN by a notable margin. Also, our HMML method outperforms the other conventional classifiers on average. We also examine the the performance of different algorithms under different training ratios $r = \{0.1, 0.3, 0.5, 0.7\}$. The results for CRAM04 dataset are shown in Figure 5. It is clear that HMML outperforms the other methods under different training ratios.

F. Considering the Effects of Sensor Sites

In this experiment, to investigate the classification performance using data captured by sensors at different physical sites, we generate training and testing dataset according to the physical sites where the UTAMS sensors are deployed. Specifically, the CRAM05 and Foreign datasets contain subsets collected from 4 different sites. We keep all the data from one site for testing and data from all the other sites for training for each dataset. The classification results are summarized in Table VI. As can be seen from this table that the proposed method performs the best on average. We can also see from Table VI that the proposed HMML method is more robust to sensors' site locations, which indicates it's potential use for real-world applications.

VI. CONCLUSION

In this paper, we have developed an effective method to jointly learn a set of heterogeneous metrics optimized for each sensor by using a multi-sensor training data in order to achieve fusion-based joint classification. The proposed method generalizes the LMNN framework which is a *state-of-the-art* single metric learning method to the setting of learning multiple metrics adapted to multiple sensors with potentially heterogeneous properties. Extensive experiments on real-world multi-sensor datasets demonstrate that the proposed method is very effective for multi-sensor fusion based classification when compared with the conventional schemes.

Acknowledgement This work is supported by U.S. Army Research Laboratory and U.S. Army Research Office under grant number W911NF-09-1-0383, and is also supported by NSF (60872145, 60903126), National High-Tech.(2009AA01Z315), Postdoctoral Science Foundation (20090451397, 201003685) and Cultivation Fund from Ministry of Education (708085) of China.

REFERENCES

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
[2] S. Boughhorbel, J. Breebaart, F. Bruekers, I. Flinsenberg, and W. ten Kate. Child-activity recognition from multi-sensor data. In *International Conference on Methods and Techniques in Behavioral Research*, 2010.
[3] D. G. Childers, D. P. Skinner, and R. C. Kemerait. The Cepstrum: A guide to processing. 65:1428–1443, 1977.
[4] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
[5] M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk. Joint manioflds for data fusion. *IEEE Trans. on Image Processing*, 19(10):2580–2594, 2010.
[6] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine Learning*, 2007.
[7] S. Engelberg and E. Tadmor. Recovery of edges from spectral data with noise-a new perspective. *SIAM Journal on Numerical Analysis*, 46(5):2620–2635, 2008.
[8] A. Fleury, M. Vacher, and N. Noury. SVM-based multi-modal classification of activities of daily living in health smart homes: sensors, algorithms and first experimental results. In *IEEE Trans. on Information Technology in Biomedicine*, volume 14, pages 274–283, 2010.
[9] I. Joliffe. *Principal compoenent analysis*. Springer-Verlag, 1986.
[10] M. Kushwaha and X. Koutsoukos. Collaborative target tracking using multiple visual features in smart camera networks. In *International Conference on Information Fusion*, 2010.
[11] S. Lee and V. Shin. Computationally efficient multisensor fusion estimation algorithms. *Journal of Dynamic Systems, Measurement, and Control*, 132(2):024503–1–024503–2, 2010.
[12] X. R. Li, Y. Zhu, and C. Han. Optimal linear estimation fusion–Part I: Unified fusion rules. *Journal of Dynamic Systems, Measurement, and Control*, 49(9):2192–2208, 2003.
[13] Z. Lu, P. Jain, and I. S. Dhillon. Geometry aware metric learning. In *International Conference on Machine Learning*, 2009.
[14] A. Martinez and A. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.
[15] C. Meesookho, S. Narayanan, and C. Raghavendra. Collaborative classification applications in sensor networks. In *Sensor array and multichannel signal processing workshop*, 2002.
[16] V. Mirelli, S. Tenney, Y. Bengio, N. Chapados, and O. Delalleau. Statistical machine learning algorithms for target classification from acoustic signature. In *Proc. MSS Battlespace Acoustic and Magnetic Sensors*, August 2009.
[17] S. Parameswaran and K. Q. Weinberger. Large margin multi-task metric learning. In *Advances in Neural Information Processing Systems*, 2010.
[18] C.-C. Shen and W.-H. Tsai. Multisensor fusion in smartphones for lifestyle monitoring. In *Int. Conf. on Body Sensor Networks*, 2010.
[19] P. Tseng and S. Yun. A coordinated gradient descent method for nonsmooth separable minimizatin. *Math. Prog.*, 117:387–423, 2009.
[20] K. R. Varshney and A. S. Willsky. Learning dimensionality-reduced classifier for information fusion. In *International Conference on Information Fusion*, 2009.
[21] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, 2006.
[22] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.
[23] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general graph framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 40–51, 2007.