INFFUS 124
21 November 2003 Disk used
ARTICLE IN PRESS
No. of Pages 10, DTD = 4.3.1

ELSEVIER

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Information Fusion xxx (2003) xxx–xxx

INFORMATION FUSION

www.elsevier.com/locate/inffus

Space efficient quantization for distributed estimation by a multi-sensor fusion system

Vasileios Megalooikonomou [a,*], Yaacov Yesha [b]

[a] Department of Computer and Information Sciences, Temple University, 1805 N. Broad Street, 303 Wachman Hall, Philadelphia, PA 19122, USA
[b] Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, MD, USA

Received 16 April 2003; received in revised form 12 November 2003; accepted 13 November 2003

Abstract

We present methods for designing quantizers for a distributed system that estimates a continuous quantity at a fusion center based on the observations of multiple sensors subject to communication constraints at the channels and to storage constraints at the fusion center. We consider the case where the observation statistics are unknown and only a training sequence is available. We propose the use of regression trees and two approaches to reduce the storage requirements. The first approach gives a direct sum estimate of the continuous quantity. The second approach provides a neural network implementation of the estimates. We study the trade-offs between storage complexity and performance using simulations. The experiments showed that the direct sum estimation approach achieves performance close to that of the unconstrained case while greatly reducing the space complexity of the fusion center. The neural network approach further improves the performance. Moreover, it provides more flexible trade-offs between space complexity and performance.

Keywords: Multi-sensor fusion; Quantization; Distributed estimation; Neural networks; Regression trees

1. Introduction

Distributed information systems consist of several separated nodes (sensors, fusion centers) observing an environment, collecting information, and making decisions or estimations based on their own observations and information that is communicated among nodes. Networks of embedded sensors are becoming increasingly important especially due to their potentially enormous impact in environmental monitoring, product quality control, defense systems, etc. New exciting technologies such as MicroElectroMechanical Systems (MEMS) [5] and Smart-Dust devices [14,28] are expected to expand the capabilities of embedded devices and networks of sensors by putting a complete sensing/communication platform inside a cubic millimeter.

Given the great technological advances and the enormous potential for applicability of sensor networks in many situations, research in data fusion in multi-sensor systems is receiving more and more attention. The main advantages of multi-sensor fusion systems [12,13,27] over single-sensor systems include the following: (a) in many applications the observations of individual sensors are incomplete, imprecise and often inconsistent so the use of multiple sensors reduces the effect of noise in measuring a quantity, (b) the use of multiple types of sensors increases accuracy in which a quantity is observed, (c) observation of a certain phenomenon may require the use of multiple sensors distributed across multiple spatial locations, (d) contextual information is very important in critical decision making. Data fusion can occur at three levels: data level, feature level, and decision level [7]. In the data level, the sensors observe the same physical phenomenon and data are directly combined. In the feature level, features are extracted from data. In the decision level, information

* Corresponding author. Tel.: +1-215-204-5774; fax: +1-215-204-5082.
E-mail addresses: vasilis@cis.temple.edu (V. Megalooikonomou), yayesha@cs.umbc.edu (Y. Yesha).

1566-2535/$ - see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/j.inffus.2003.11.002

about an entity's attributes is extracted from each sensor.

Here, we focus on fusion at the data level. We consider a distributed system of a fusion center with a number of remote sensors. The fusion center makes decisions that are based on data collected by remote sensors and transmitted to the center. Depending on the application, the decision making at the fusion center can be in the form of binary hypothesis testing (e.g., radar detection) [1,6,20,24,25], or estimation (e.g., target tracking using multiple radars) [8,11,16,17,29]. Although here we consider the case of a single fusion center, the work has applicability to hop-by-hop communication networks with many fusion centers where the fusion center considered can be a cluster-head sensor of its own cluster. Especially in wireless communication networks, in order to conserve battery resources and/or communicate around obstacles, short-range hop-by-hop communication is often preferred over direct long-range communication to a destination. In this case, the design of the multiple-hop network can be performed using the proposed approach by designing separately in a cascading manner each cluster of sensors starting from the leaves of the tree and proceeding towards the root.

We concentrate on distributed systems that perform estimation of a certain quantity at the fusion center using the observations of the sensors. Examples of such quantities or conditions are: temperature, humidity, noise levels, movement of objects (such as vehicles, equipment, robotic devices), mechanical stress levels, etc. If these observations are directly available at the fusion center then the problem is termed *centralized*, and it can be carried out by more traditional methods of detection and estimation theory [26]. Here, we consider the *distributed* problem where the observations are not directly available to the fusion center. Rather, they are collected at the fusion center through communication channels with capacity constraints [11,16,17,20]. This is the case in wireless sensor networks where sensors have wireless communication capabilities and may be battery constrained and operate under power conservation conditions. The capacity constraints on communication along with the storage constraints on the fusion center suggest very challenging problems.

The scheme considered serves as a model for many applications from environmental monitoring, to homeland security and quality control and from seismology to meteorology and medicine. The goal is to minimize the expected error in estimating a continuous quantity. Due to constraints on the communication lines imposed by power limitations (battery constrained) and wireless capabilities, there are several restrictions on the model we consider: the sensors cannot communicate with each other and there is no feedback from the fusion center back to the sensors. Due to limitation in communication bandwidth the observations are compressed (quantized). The estimation is achieved via compressed information. We assume error free communication channels and fixed length coding for the transmission. We also assume that the observation statistics, i.e., the joint probability density function, is unknown.

The problem we are considering is defined as follows: For a distributed system with k sensors, find, for each sensor, a mapping from the observation space to codewords (of a certain number of bits given by the capacity constraints), and find a fusion center function that maps a vector of k codewords to an estimate vector for the unobserved quantities, so that the mean of the square of the Euclidean norm of the estimation error is minimized. The representation of the fusion center function may take into account the storage constraints at the fusion center. There is a joint probability distribution of all observations and unobserved quantities. However, since this distribution is unknown, the design of the system is based on a training set and the mean squared error is computed based on a test set. Although the number of sensors, k, can be in general arbitrary, here we consider the two-sensor case (see Fig. 1) since the method for this case can be easily extended to the more general case.

We assume that a training set is available. Training data can usually be obtained, with some additional cost, in the collection process. For example, in the case of remote object tracking, in addition to the estimates at the sensors the actual location of the object can be available by other means (e.g., the object moves on a predetermined path with a known speed to collect data for training before the design phase). This is similar to a calibration procedure.

The problem of quantizer design for a distributed estimation system in the case of unknown observation statistics was considered by Megalooikonomou and

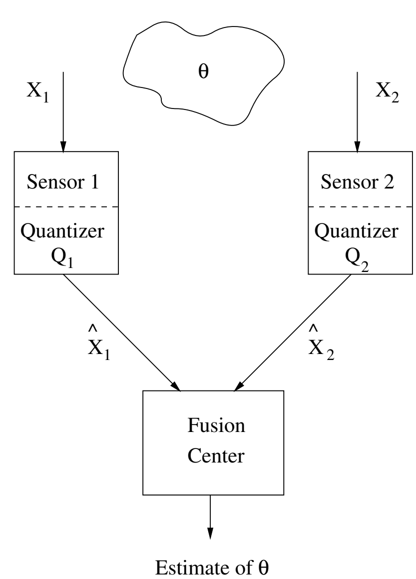

Fig. 1. A distributed estimation system with two sensors.

Yesha [21–23]. The same problem in the case of a known probability model was considered by Lam and Reibman [16,17]. Longo, et al. [20] considered the design of the peripheral encoders for a decentralized hypothesis testing network under communication constraints in the case where the joint distribution is known. Ephraim and Gray [8] and Ayanoglou [2] studied quantization for estimation for the single sensor case. However, storage complexity issues for the fusion center were not addressed in those papers.

The methods that have been proposed for designing quantizers for distributed estimation use a simple table for the fusion center that is indexed by the codewords of the quantizers. Storage requirements of this fusion center table can be a problem for a large number of sensors and/or a large number of codewords for each sensor (space for the fusion center table is exponential in the number of sensors). Gubner [11] considers the problem of quantizer design for this system subject not only to communication constraints but also to computation constraints at the fusion center. Gubner, though, assumes that the probability model is known. The computational capabilities of the fusion center are constrained to direct sum estimation of the continuous quantity. As a result, his algorithm uses only bivariate joint distributions. We design methods that deal with the case of unknown distribution.

In this paper we consider the problem of quantizer design subject to both communication constraints and storage constraints at the fusion center in the case where the joint probability model is unknown and one must rely on a training set. To deal with the unknown joint probability model we use a regression tree approach for designing the quantizers. We propose two approaches that use reduced storage representations of the fusion center table. The first approach gives a direct sum estimation of the continuous quantity. The second approach uses a neural network for the representation of the reconstructed values at the fusion center.

2. Background

Let θ be the unobservable continuous (in values) quantity that the fusion center tries to estimate. Let X_1^q and X_2^r be the random observation vectors at the two sensors, where q and r denote the number of elements in each vector. We use the vector notation X_k^p as a shorthand for $(X_k[1], X_k[2], \ldots, X_k[p])$. Let $\mathscr{T} = \{(X_1^q, X_2^r)^{(t)}, \theta^{(t)}; t = 1, \ldots, M\}$ be a training set of size M that represents the statistics of the source. Note that when referring to the random observation vectors present in the training set we use the notation $X_k^{p,t}$ where as before p and k denote, respectively, the number of elements in the vector and the particular sensor, while t denotes the particular sample from the training set $\mathscr{T}$.

Let Q_k be the quantizer for sensor k, and $Q_k(X_k^{p,t})$ or, in other terms, $\hat{X}_k^{p,t}$ (denoting that this is the quantized version of $X_k^{p,t}$) be the codeword for the observation $X_k^{p,t}$ that is transmitted to the fusion center. The task of the fusion center is to estimate the unobserved quantity θ based on the $\hat{X}_k^{p,t}$ it receives. Note that the codeword received is the same as the codeword transmitted due to the assumption of error-free communication channels. Let h be the function of the fusion center that gives the estimate of θ and

$$P_{Q_1} = \{U_i; i = 1, \ldots, N\} \tag{1}$$

and

$$P_{Q_2} = \{V_j; j = 1, \ldots, L\}, \tag{2}$$

be the sets of partition regions for quantizers Q_1 and Q_2, respectively. The transmitted values, $\hat{X}_1^{q,t}$ and $\hat{X}_2^{r,t}$ from the sensors to the fusion center are given by: $\hat{X}_1^{q,t} = \sum_{i=1}^{N}(i-1)I_{U_i}(X_1^{q,t})$ and $\hat{X}_2^{r,t} = \sum_{j=1}^{L}(j-1)I_{V_j}(X_2^{r,t})$, where $I_A(x)$ denotes the indicator function of a set $A \subset \Re^d$ of dimension d, i.e., $I_A(x) = 1$ if x is in A and $I_A(x) = 0$ otherwise. Then the output of the distributed estimation system is $h(\hat{X}_1^{q,t}, \hat{X}_2^{r,t})$ or in other words $h(Q_1(X_1^{q,t}), Q_2(X_2^{r,t}))$. The fusion center h has the following value for each pair of codewords i, j:

$$h(i,j) = \frac{1}{|\mathscr{R}_{i,j}|} \sum_{t:(X_1^q, X_2^r)^{(t)} \in \mathscr{R}_{i,j}} \theta^{(t)} \tag{3}$$

where $\mathscr{R}_{i,j} = \{(X_1^q, X_2^r)^{(t)} : X_1^{q,t} \in U_i, X_2^{r,t} \in V_j\}$ is a subset of the training set. We consider the mean-squared error (MSE) distortion function. The objective is to find Q_1, Q_2, and h such that the error expression below is minimized

$$\text{Error} = \frac{1}{M} \sum_{t=1}^{M} \left(\theta^{(t)} - h\left(\hat{X}_1^{q,t}, \hat{X}_2^{r,t}\right)\right)^2 \tag{4}$$

The numbers N, L of partition regions for quantizers Q_1 and Q_2 respectively are provided so that the capacity constraints on the communication channels are satisfied.

In the case where the joint distribution $p(x_1, x_2, \theta)$ (i.e., the observation statistics) is known and continuous, necessary conditions for optimal Q_1, Q_2, and h for the MSE distortion function are given by Lam and Reibman [17]. These conditions are not sufficient. However, their joint solution leads to an estimation error that converges. It is widely believed that this solution is indeed locally optimal, although no general theoretical derivation of this result has ever been obtained [9]. In order to find the solution, the Cyclic Generalized Lloyd's Algorithm (CGLA) proposed by Longo, et al. [20] in the framework of decentralized hypothesis testing under capacity constraints and for a known joint distribution is used [11,16,17,20]. The CGLA is a variation of the Generalized Lloyd Algorithm (GLA) [10,18,19], a

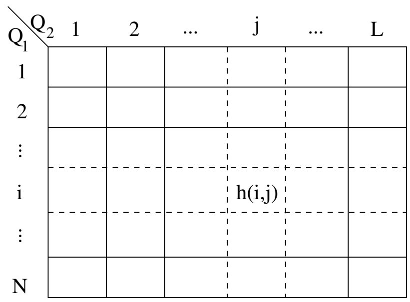

Fig. 2. A fusion center table for distributed estimation with two sensors.

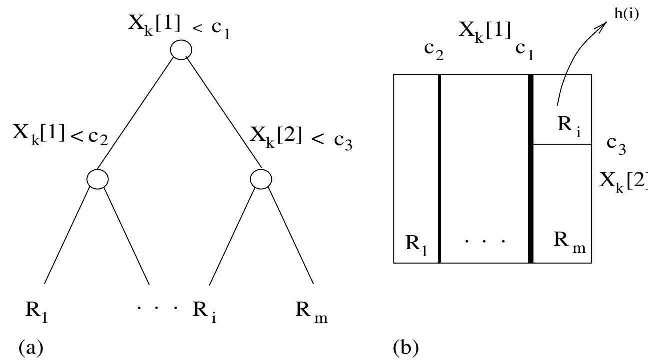

Fig. 3. A regression tree of two variables (a) and its partitions (b).

well-established methodology for designing single quantizers when the aim is to minimize the distortion [10,18,19]. It starts with an initial guess of quantizers and fusion center and it iteratively improves them by finding the optimal component given the others.

The methods proposed for designing quantizers for distributed estimation use a simple table for the fusion center that is indexed by the codewords of the quantizers (see Fig. 2 for an example of a fusion center table for two quantizers). The amount of space required for storing this fusion center table is exponential in the number of sensors. For example, a fusion center table for 10 sensors and just 8 codewords per sensor should have 8^{10} elements! Here, we propose methods for reducing the space requirements of the fusion center while at the same time satisfying the capacity constraints of the communication channels and minimizing the estimation error.

In the following sections we introduce regression trees and present background information on how to use them to design quantizers.

2.1. Regression trees

Regression analysis is the generic term revolving around the construction of a predictor from a training set. Breiman et al. [4] describe tree structured predictors (BFOS regression trees) that are formed by iteratively splitting subsets (nodes) of the training set into descendant disjoint subsets, beginning with the training set itself, in order to maximize the decrease in the mean prediction error. In each terminal node (leaf) the predicted response value is either constant (where the tree can be thought of as a histogram estimate of the regression surface) or some approximating function.

The main issues in designing regression trees are the assignment of a value to every terminal node, the selection of good splits (queries), and the stop splitting rules. However, in order to grow trees of the right size, instead of attempting to stop the splitting at the right set of terminal nodes, one may continue the splitting until the expected prediction error is below a certain threshold (resulting in a large tree), and then selectively prune this large tree by recombining leaves that are siblings.

Regression trees [4] are decision trees with queries of the form $X_k[l] < c_j$ (for an observation variable $X_k[l]$ and a constant c_j) where each leaf R_i is labeled by an estimation value $h(i)$ which is generally constant. See Fig. 3(a) for a regression tree on two variables, $X_k[1]$ and $X_k[2]$, and Fig. 3(b) for the representation of its partitions. For observations of dimension d the leaves of the regression tree correspond to d-dimensional rectangles. All splits are on single variables so they are perpendicular to the coordinate axes. The regression tree is grown by introducing a split at a time. The basic operation that finds the next split is as follows: At each node the tree algorithm searches through the variables one by one. For each variable it finds the split that results in the greatest reduction in prediction error. Then it compares the best single variable splits and selects the best among them for the split at this node. Finally, it splits the node for which the greatest reduction of the prediction error was noticed.

2.2. Designing quantizers using regression trees

In order to deal with the problem of unknown joint distribution we design the quantizers using the regression tree approach proposed by Megalooikonomou and Yesha [21]. In this section, having introduced regression trees, we present details on the design of quantizers.

As mentioned earlier the regression trees are formed by iteratively splitting subsets of the training set into decendant disjoint subsets in order to reduce the estimation error. For sensor k the next split is chosen (considering all the variables $X_k[1], \dots, X_k[p]$ and all the values of these variables) so that the error in the estimation of the quantity θ, given by Eq. (4), is minimized. The tree growing is cooperative since the estimation error depends on the existing rectangles of both trees. In order to grow trees of the right size, pruning is also involved in the growing procedure. The pruning algorithm that is used, which recombines leaves that are siblings, is the Recursive Optimal Pruning Algorithm (ROPA) proposed by Kiang et al. [15]. The purpose of pruning the original regression trees in the case of fixed rate quantization is to get a subtree with a given number of leaves and with estimation error that is as small as possible. When one tree is pruned given the other tree, the risk of every node (i.e., the expected estimation error of the subtree that has as root that node [4]) in the tree is calculated taking into account the other tree.

After building a regression tree for each one of the sensors the rate is reduced using a labeling technique that combines rectangles into the required number of partition regions assigning the same codeword (label) to the rectangles of the same region. The rectangles are labeled using s-CGLA (set-CGLA), an algorithm that considers together groups of training samples and is related to the Cyclic Generalized Lloyd Algorithm (CGLA) [20]. A variation of s-CGLA is the lh-s-CGLA (lookahead-set CGLA) that changes the fusion center temporarily whenever there is a decision that has to be made in order to calculate the effect of every possible change and also keeps the fusion center table updated all the time.

Let n_k be the number of codewords and $m_k \geqslant n_k$ be the number of leaves for quantizer k. Let also $l(r)$ be the label of a specific rectangle r. Given the partition regions P_{Q_1} and P_{Q_2}, for X_1^q and X_2^r respectively, the optimal fusion center h is given by:

$$h(m,n) = \frac{1}{|\mathscr{R}'_{m,n}|} \sum_{t:(X_1^q,X_2^r)^{(t)} \in \mathscr{R}'_{m,n}} \theta^{(t)} \tag{5}$$

where $\mathscr{R}'_{m,n} = \{(X_1^q, X_2^r)^{(t)} : l(r(X_1^{q,t})) = m, l(r(X_2^{r,t})) = n\}$ is a subset of the training set. The estimation error, err_i, contributed by the subset, $\mathscr{R}''_i = \{(X_1^q, X_2^r)^{(t)} : r(X_1^{q,t}) = i\}$, of the training set (based on X_1^q) is given by:

$$\mathrm{err}_i = \frac{1}{|\mathscr{R}''_i|} \sum_{t:(X_1^q,X_2^r)^{(t)} \in \mathscr{R}''_i} \left(\theta^{(t)} - h(l(i), l(r(X_2^{r,t})))\right)^2 \tag{6}$$

The total estimation error is then given by:

$$\mathrm{Error} = \sum_{i:0\ldots m_1-1} \mathrm{err}_i \tag{7}$$

The estimation error can also be expressed using a similar formula that includes the corresponding subsets of the training set based on X_2^r. The main component of lh-s-CGLA performs the following for each sensor k [21] until the reduction on the estimation error given by Eq. (7) is less than a given threshold:

for each rectangle i from 0 to $m_k - 1$ **do**
  for each label j from 0 to $n_k - 1$ **do**
    $l(i) \leftarrow j$
    calculate h (Eq. (5)), $\mathrm{err}_i[j]$ (Eq. (6))
  $l(i) \leftarrow \arg\min_{j:0\ldots(n_k-1)}(\mathrm{err}_i[j])$
  calculate h (Eq. (5))

The breakpoint initialization method [23] is used to initialize the labels of the rectangles.

3. Methods for reducing the storage requirements of the fusion center

Here, we introduce two fusion strategies with a reduced space complexity for the fusion center. The first approach gives a direct sum estimation of the continuous quantity. The second approach uses a neural network representation of the reconstructed values at the fusion center. We also propose a combination of the two approaches.

3.1. Direct sum estimation of the continuous quantity

Recall that the transmitted value $\hat{X}_1^q = i - 1$ if and only if the observation X_1^q is in the partition region U_i, $i = 1, \ldots, N$ of quantizer Q_1, and similarly, the transmitted value $\hat{X}_2^r = j - 1$ if and only if the observation X_2^r is in the partition region V_j, $j = 1, \ldots, L$ of quantizer Q_2. Constraining the storage requirements of the fusion center using the direct sum estimation approach we require (as in the approach proposed by Gubner [11]) that the estimation of the quantity θ for a certain training point t is:

$$\hat{\theta}^{(t)} = \sum_{i=1}^{N} a_i I_{U_i}\left(X_1^{q(t)}\right) + \sum_{j=1}^{L} b_j I_{V_j}\left(X_2^{r(t)}\right) \tag{8}$$

where $a_i : i = 1, \ldots, N$ and $b_j : j = 1, \ldots, L$ are the parameters of the fusion center that we are trying to find, N, L, are the numbers of partition regions for quantizers Q_1 and Q_2 respectively (see Eqs. (1) and (2)), and $I_A(\cdot)$ is the indicator function of a set A as defined earlier. If the partition regions of the sensors are fixed, choosing the a_i's, b_j's that minimize the error

$$\mathrm{Error}_{\mathrm{lin}} = \frac{1}{M} \sum_{t=1}^{M} \left(\theta^{(t)} - \hat{\theta}^{(t)}\right)^2 \tag{9}$$

is a linear estimation problem whose solution is given by the normal equations (also used by Gubner [11]). We use an iterative solution to the normal equations. The parameters a_i, $i : 1, \ldots, N$ are given as follows:

$$a_i = \frac{1}{|\mathscr{R}_i|} \left(\sum_{j=1}^{L} \sum_{t:(X_1^q,X_2^r)^{(t)} \in \mathscr{R}_{i,j}} \theta^{(t)} - \sum_{j=1}^{L} |\mathscr{R}_{i,j}| b_j \right) \tag{10}$$

where $\mathscr{R}_i = \{(X_1^q, X_2^r)^{(t)} : X_1^{q,t} \in U_i\}$, and $\mathscr{R}_{i,j} = \{(X_1^q, X_2^r)^{(t)} : X_1^{q,t} \in U_i, X_2^{r,t} \in V_j\}$ are subsets of the training set. A similar formula is derived for the parameters b_j, $j : 1, \ldots, L$. We calculate the parameters a_i, $i : 1, \ldots, N$ and b_j, $j : 1, \ldots, L$ iteratively until the reduction on the estimation error given by Eq. (9) is less than a given threshold. We initialize the parameters b_j, $j: 1, \ldots, L$ with 0. This iterative method solves the normal equations using a training set and converges to the actual solutions given by analytical methods. It is actually the Gauss–Seidel (successive displacements) method where each new component is immediately used in the calculation of the next component.

When the partition regions of one sensor along with the fusion center parameters are fixed we partition the space of the other quantizer using a variation of the methods for growing and pruning the regression trees that were used in the case of an unrestricted fusion center [21]. This variation of the methods is used because of the restrictions on the storage requirements of the fusion center. The estimation error is now calculated using Eq. (9). Also, when we build the regression tree for X_1^q taking into account the partitions of the quantizer for X_2^r along with their associated parameters b_j, $j: 1, \ldots, L$ we use the quantities $\theta^{(t)} - b_j$ (where $X_2^{r(t)}$ is in partition region V_j) instead of $\theta^{(t)}$ in the methods that try to find the best split on X_1^q. A similar variation is used when we build the regression tree for X_2^r given the regression tree for X_1^q. It can be proved that minimizing the expected squared estimation error based on the above quantities is equivalent to minimizing the expected squared estimation error based on the direct sum estimate. The proof is omitted.

In the case of direct sum estimation, finding the best splits for one tree, when the other tree and the coefficients that correspond to it are fixed, is feasible. We build the quantizers taking into account the special form that the fusion center function takes in order to satisfy the storage constraints.

3.2. Neural network representation of the reconstructed values

By partially relaxing the space requirements of the fusion center we can further improve the performance achieved by the direct sum estimation of the quantity using a neural network representation of the fusion center table (i.e., of the reconstructed values given by Eq. (3)). The neural network output (after training) approximates the reconstructed values of the fusion center. Simulations show (see Table 2) that with enough parameters (that correspond to weights and biases of neurons) the neural network can achieve better performance than the one given by the direct sum estimation method. Moreover, we can have control on the number of parameters that need to be stored in order to approximate the fusion center table h.

The neural network that we use is a two-layer feedforward network and the learning rule is backpropagation with momentum and adaptive learning rate. The momentum method decreases the probability that the network will get stuck in a shallow minimum in the error surface and helps decrease training times. Adaptive learning rate decreases training time by keeping the learning rate reasonably high while insuring stability. For the first layer we use a hyperbolic tangent transfer function and for the second layer we use a linear transfer function. This kind of networks has been proven capable of approximating any function with a finite number of discontinuities with arbitrary accuracy [3]. By varying the number of neurons, we can achieve various tradeoffs of the complexity of the fusion center representation and the performance of our system.

Let S_1 be the number of neurons in the first layer of the neural network. We use only one neuron for the second layer (the output layer). We use the notation $c_k : k = 1, \ldots, M$ for the elements of the weight and bias matrices that are, in other words, the parameters of the neural network. Let I be the number of inputs of the neural network. These correspond to the codewords of the quantizers. We use the unary representation of the codewords with each input corresponding to a codeword being present or absent. Therefore, the number of inputs for the neural network is equal to the total number of codewords from all the quantizers. Then the number of parameters, M, used for the description of the two-layer neural network is

$$S_1(I + 2) + 1. \tag{11}$$

The number of parameters for the weights and the biases of the neural network depends on the representation of the input.

We use the Nguyen–Widrow initial conditions for the weights and the biases in order to reduce even further the training time. For the training we can use either the quantities, $\theta^{(t)}$, or the corresponding fusion table entry, $h^{(t)}$, given by:

$$h^{(t)} = \sum_{i=1}^{N} \sum_{j=1}^{L} h(i,j) I_{U_i}\left(X_1^{q(t)}\right) I_{V_j}\left(X_2^{r(t)}\right) \tag{12}$$

for every point t of the training set, where $h(i,j)$ is given by Eq. (3). This is because the corresponding two expressions for the estimation error differ by a quantity that depends only on the training set (the proof is easy and is omitted). However, the learning process is faster in the second case because of the variation of the target values for the same input in the first case.

If $f(i,j)$ is the output of the neural network for inputs i, j after the training, then the approximation of the fusion center table entry $h^{(t)}$, $\hat{h}^{(t)}$, is given by:

$$\hat{h}^{(t)} = \sum_{i=1}^{N} \sum_{j=1}^{L} f(i,j) I_{U_i}\left(X_1^{q(t)}\right) I_{V_j}\left(X_2^{r(t)}\right) \tag{13}$$

for every point t of the training set. Then the total estimation error can be expressed as follows:

$$\text{Error}_{\text{NN}} = \frac{1}{M}\sum_{t=1}^{M}\left(\theta^{(t)} - \hat{h}^{(t)}\right)^2 \tag{14}$$

In order to improve the performance of the system further we allow changes of the labels of the quantizer rectangles taking into account the last form of the fusion center that now depends on the parameters, c's, of the neural network. We apply the lh-s-CGLA in order to decide about the best label of a rectangle. The algorithm, *design-fc-q*, that iteratively trains the neural network used for the fusion center table and possibly alters the quantizers by assigning new labels to the rectangles, based on the new fusion center, is described below. The variable l is the iteration counter, h^l the fusion center at iteration l, and c^l's the neural network parameters at iteration l.

design-fc-q
1. $l \leftarrow 1$.
2. run lh-s-CGLA to assign better labels to the rectangles of the quantizers.
3. calculate h^l, c^l's (by constructing a neural network to approximate the fusion center h^l), and Error_{NN} (Eq. (14)).
4. **if** the reduction of the estimation error is less than a given threshold, **then** stop, **else** $l \leftarrow l+1$ and **goto** step 2.

Notice that an approach that, in addition to changing the labels, attempts to change also the split points of the quantizers, is not computationally feasible due to the time involved in training the neural network for every possible change of the split points.

3.3. Combining direct sum estimation with neural networks

An alternative solution to the neural network representation of the reconstructed values given by Eq. (3) is the use of a neural network representation for the residual of the reconstructed values in addition to the parameters $a_i : i = 1,\dots,N$ and $b_j : j = 1,\dots,L$ of the direct sum estimation method. In other words, in this combined approach, we take into account the direct sum estimation given by Eq. (8) when training the neural network. For the training we can use either the residual r_1, $r_1^{(t)} = \theta^{(t)} - \hat{\theta}^{(t)}$, or the residual r_2 given by:

$$r_2^{(t)} = h^{(t)} - \hat{\theta}^{(t)} \tag{15}$$

for every point t of the training set, where $h^{(t)}$ is the corresponding fusion table entry given by Eq. (12) and $\hat{\theta}^{(t)}$ is the estimation of $\theta^{(t)}$ by the direct sum estimation system. Either quantity can be used in the training because the objective functions in the two cases differ by a constant that depends only on the training set.

In order to achieve even better performance given the total number of parameters that we can use we propose an iterative approach for finding the parameters of the direct sum estimation method (a's and b's) and those of the neural network (c's). If the approximation $\hat{\theta}'$ of the residual r_2 by the neural network is given as:

$$\hat{\theta}'^{(t)} = \sum_{i=1}^{N}\sum_{j=1}^{L} f(i,j) I_{U_i}\left(X_1^{q(t)}\right) I_{V_j}\left(X_2^{r(t)}\right) \tag{16}$$

for every point t of the training set, the total estimation error can be expressed as follows:

$$\text{Error}_{\text{tot}} = \frac{1}{M}\sum_{t=1}^{M}\left(\theta^{(t)} - \left(\hat{\theta}^{(t)} + \hat{\theta}'^{(t)}\right)\right)^2. \tag{17}$$

In this case, we use the following iterative algorithm, *design-fc*, to calculate the parameters a's, b's, and c's of the fusion center:

design-fc
1. $l \leftarrow 1$, $\hat{\theta}' \leftarrow 0$.
2. calculate a^l's (Eq. (10)), b^l's iteratively to approximate the residual $\theta - \hat{\theta}'$.
3. calculate $\hat{\theta}$ (Eq. (8)), c^l's (by constructing a neural network to approximate r_2 (Eq. (15))), $\hat{\theta}'$ (Eq. (16)), and $\text{Error}_{\text{tot}}$ (Eq. (17)).
4. **if** the reduction on the estimation error is less than a given threshold, **then** stop, **else** $l \leftarrow l+1$ and **goto** step 2.

In order to further improve the performance of this system we apply the lh-s-CGLA that allows changes of the labels of the quantizer rectangles taking into account the last form of the fusion center that now depends on the parameters of the direct sum estimation system (a's and b's) and on the parameters of the neural network (c's).

4. Experimental investigations and discussion

In the experiments we consider the case where the observations at the quantizers are scalar quantities of the form:

$$x_k = \theta + n_k, \quad k = 1,2 \tag{18}$$

where the noises n_k at the sensors are Gaussian distributed with correlation coefficient ρ and marginal distributions $N(0,\sigma_n^2)$, where σ_n^2 is the variance of the noises. The continuous quantity θ has Gaussian distribution $N(0,1)$ and is independent of the noises n_k, $k = 1,2$. The quantizers are designed using a training set $\mathcal{T}$ of 5000 samples and are tested on a test set $\mathcal{T}'$ of 5000 samples that is independent of $\mathcal{T}$ although it is constructed the same way as $\mathcal{T}$. The results that we report here are on the test set $\mathcal{T}'$. In all experiments we use the breakpoint initialization of labels and the value 0.005 for the error

Table 1
Comparison of the performance of the unconstrained (full fusion center table) and the direct sum estimation approach

σ_n^2	bp_init, 16 leaves, (8, 8) labels					
	$\rho = 0$		$\rho = 0.5$		$\rho = 0.85$	
	No-constr. 64 params	Dir. sum. est. 16 params	No-constr. 64 params	Dir. sum. est. 16 params	No-constr. 64 params	Dir. sum. est. 16 params
0.005	0.0094	0.0172	0.0093	0.0173	0.0104	0.0216
0.010	0.0135	0.0221	0.0145	0.0226	0.0250	0.0288
0.050	0.0381	0.0466	0.0493	0.0582	0.0554	0.0627
0.100	0.0674	0.0762	0.0878	0.0984	0.0976	0.1198
0.150	0.0886	0.0982	0.1233	0.1303	0.1371	0.1505
0.200	0.1154	0.1288	0.1540	0.1703	0.1816	0.2038
0.300	0.1573	0.1685	0.2013	0.2210	0.2436	0.2611
0.400	0.1897	0.2053	0.2550	0.2701	0.2998	0.3178
0.500	0.2209	0.2518	0.2945	0.3213	0.3470	0.3698
0.600	0.2521	0.2702	0.3383	0.3625	0.3901	0.4100
0.700	0.2849	0.3030	0.3730	0.4081	0.4260	0.4509
0.800	0.3126	0.3295	0.3989	0.4157	0.4643	0.4868
0.900	0.3410	0.3588	0.4234	0.4502	0.4923	0.5109
1.000	0.3621	0.3796	0.4593	0.4835	0.5126	0.5523

threshold. We also use 10,000 epochs to train the neural network. Here we present results from experiments with 2 sensors and 8 partition regions for each quantizer.

In Table 1 we compare the estimation error of the unconstrained method (without restriction on the fusion center) with that of the direct sum estimation method that constrains the capabilities of the fusion center using Eq. (8). We assume breakpoint initialization of labels and we present results for several values of σ_n^2 and for $\rho = 0,\ 0.5,\ 0.85$. Despite the great reduction on the number of parameters used in the direct sum estimation method its performance is close to that of the unconstrained method.

In Fig. 4 we present the performance of the direct sum estimation method in the case where we do not take into account the restrictions on the fusion center when building the regression trees and we only apply the procedure that calculates the parameters of the fusion center as the last step (a). We also present the performance in the case where the procedure that builds the quantizers takes into account the special form that the fusion center function takes in order to satisfy the storage constraints (b). We present the results for $\rho = 0.85$. The difference in the estimation error is similar for $\rho = 0,\ 0.5$. As expected, it is better to build the quantizers taking into account, during the whole process, the restricted form of the fusion center.

In Table 2 we compare the performance of the direct sum estimation system with that of the neural network with 3 and 4 neurons. The number of parameters used by each system is also shown. It is clear that the neural network approach provides flexible trade-offs between storage complexity of the fusion center (i.e., number of neurons used) and performance of the quantizers. The results that we report here for the neural network are

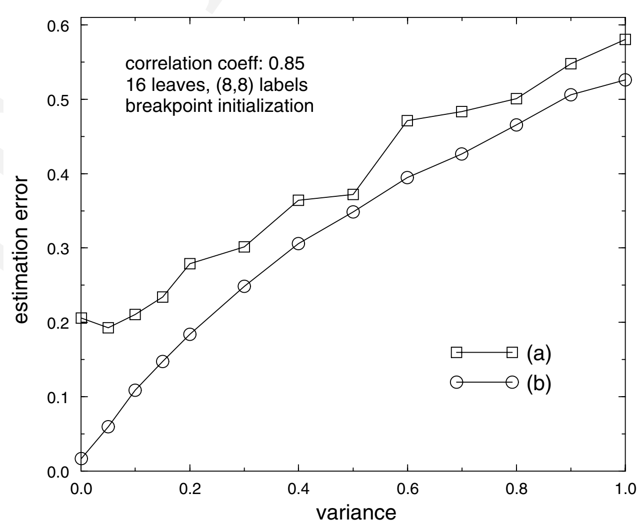

Fig. 4. Performance comparison of the direct sum estimation method in the case (a) where we build the quantizers without taking into account in the building process the special form of the fusion center function and in the case, (b) where we take it into account during the whole building process.

obtained after we apply the *design-fc-q* algorithm that iteratively trains the neural network for the fusion center and possibly alters the quantizers by assigning new labels to the rectangles of their regression trees. Table 3 presents the improvement of the *design-fc-q* algorithm over the one-iteration approach that builds a neural network of 4 neurons and stops.

Finally, in Table 4 we compare the direct sum estimation system with the one that in addition to the a's and b's, uses a neural network representation of the residual of the reconstructed values. We iteratively improve the direct sum estimation system and the neural

Table 2
Comparison of estimation errors (MSE) between a direct sum estimation and a neural network approximation of the fusion center table

σ_n^2	bp_init, $\rho = 0.85$, 16 leaves, (8, 8) labels		
	Dir. sum. est. 16 params	3 neurons 37 params	4 neurons 55 params
0.005	0.0216	0.0207	0.0106
0.010	0.0288	0.0270	0.0261
0.050	0.0627	0.0588	0.0571
0.100	0.1198	0.1078	0.0984
0.150	0.1505	0.1381	0.1370
0.200	0.2038	0.1880	0.1819
0.300	0.2611	0.2501	0.2481
0.400	0.3178	0.2999	0.2996
0.500	0.3698	0.3505	0.3474
0.600	0.4100	0.3952	0.3920
0.700	0.4509	0.4307	0.4261
0.800	0.4868	0.4650	0.4637
0.900	0.5109	0.4982	0.4939
1.000	0.5523	0.5269	0.5144

Table 3
Comparison of estimation error (MSE) for the neural network approach—one iteration versus *design-fc-q*

σ_n^2	bp_init, $\rho = 0.85$, 16 leaves, (8, 8) labels	
	One iteration	*Design-fc-q*
0.005	0.0132	0.0106
0.010	0.0287	0.0261
0.050	0.0571	0.0571
0.100	0.1079	0.0984
0.150	0.1466	0.1370
0.200	0.1904	0.1819
0.300	0.2489	0.2481
0.400	0.3031	0.2996
0.500	0.3509	0.3474
0.600	0.4034	0.3920
0.700	0.4279	0.4261
0.800	0.4725	0.4637
0.900	0.4953	0.4939
1.000	0.5306	0.5144

network using the *design-fc* algorithm. We report the case of a neural network with one neuron (hyperbolic tangent transfer function), two neurons (one in the hidden layer, one in the output layer) and three neurons (two in the hidden layer, one in the output layer). We also report the total number of parameters used in each case. We present the results for $\rho = 0.85$. As expected, the performance of this system is similar to the one that uses a neural network for the representation of the fusion center table instead of a neural network for the representation of the residual table in addition to the direct sum estimation of the continuous quantity.

5. Conclusions

In this paper we have addressed the problem of designing efficient quantizers for a multi-sensor fusion system that performs estimation, where the efficiency is in terms of space complexity of the fusion center. In our system, quantization is used to meet the communication constraints between the sensors and the fusion center. Previous work on this problem assumed partial knowledge of the data statistics. However, here, we considered the case of unknown data statistics, and the system design was accordingly based on training sets.

To reduce the fusion center space requirements we proposed two approximations of the estimation rule: a direct sum estimation and a neural network implementation of the estimates. In addition, we considered a combination of the two approaches. We performed numerical investigations to quantify the estimation error of the proposed approaches. Experiments demonstrated that the performance loss observed for the direct sum estimation approach was small while the space complexity was greatly reduced. The neural network approach provided more flexible trade-offs between

Table 4
Performance improvement (in terms of MSE) when combining the neural network with the direct sum estimation approach

σ_n^2	bp_init, $\rho = 0.85$, 16 leaves, (8, 8) labels			
	Dir. sum. est. (dse) 16 params	dse + 1 neuron 33 params	dse + 2 neurons 35 params	dse + 3 neurons 53 params
0.005	0.0216	0.0217	0.0197	0.0145
0.010	0.0288	0.0268	0.0265	0.0253
0.050	0.0627	0.0589	0.0580	0.0561
0.100	0.1198	0.1079	0.1061	0.0987
0.150	0.1505	0.1453	0.1422	0.1381
0.200	0.2038	0.1831	0.1831	0.1828
0.300	0.2611	0.2461	0.2460	0.2460
0.400	0.3178	0.3031	0.3026	0.3011
0.500	0.3698	0.3469	0.3466	0.3465
0.600	0.4100	0.3916	0.3913	0.3904
0.700	0.4509	0.4240	0.4238	0.4238
0.800	0.4868	0.4625	0.4624	0.4615
0.900	0.5109	0.5029	0.5005	0.4972
1.000	0.5523	0.5222	0.5214	0.5212

storage complexity of the fusion center and performance of the quantizers with the performance loss becoming even smaller than that of the direct estimation for slightly increased space requirements. Another important observation, which was expected but our experiments confirmed, was that one can build better quantizers by taking into account, throughout the design process, the restricted form of the fusion center rather than by imposing the restriction at the end. The main contribution of this work was the introduction of two fusion strategies and the modification of previously proposed regression-tree techniques to reduce the space complexity of the fusion center.

Acknowledgement

The authors are grateful to the anonymous referees for very helpful comments.

References

[1] R. Alshwede, I. Csiszár, Hypothesis testing with communication constraints, IEEE Transactions on Information Theory IT-32 (1986) 533–542.
[2] E. Ayanoglu, On optimal quantization of noisy sources, IEEE Transactions on Information Theory IT-36 (6) (1990) 1450–1452.
[3] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press Inc., New York, USA, 1995.
[4] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth Inc, Belmont, CA, USA, 1984.
[5] L.R. Carley, G.R. Ganger, D.F. Nagle, MEMS-based integrated-circuit mass-storage systems, Communications of the ACM 43 (11) (2000) 73–80.
[6] Z. Chair, P.K. Varshney, Optimal data fusion in multiple sensor detection systems, IEEE Transactions on Aerospace and Electronic Systems AES-22 (1) (1986) 98–101.
[7] J.J. Clarke, A.L. Yuille, Data Fusion for Sensory Information Processing Systems, Kluwer Academic Publishers, Dordrecht, 1990.
[8] Y. Ephraim, R.M. Gray, A unified approach for encoding clean and noisy sources by means of waveform and autoregressive model vector quantization, IEEE Transactions on Information Theory IT-34 (4) (1988) 826–834.
[9] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992.
[10] R.M. Gray, Vector quantization, IEEE ASSP Magazine 1 (1984) 4–29.
[11] J.A. Gubner, Distributed estimation and quantization, IEEE Transactions on Information Theory IT-39 (4) (1993) 1456–1459.
[12] D.L. Hall, J. Llinas, An introduction to multisensor data fusion, Proceedings of the IEEE 85 (1) (1997) 6–23.
[13] D.L. Hall, J. Llinas, Handbook of Multisensor Data Fusion, CRC Press, 2001.
[14] J. Kahn, R. Katz, K. Pister, Emerging challenges: mobile networking for smart dust, Journal of Communications and Networks 2 (3) (2000) 188–196.
[15] S.-Z. Kiang, R.L. Baker, G.J. Sullivan, C.-Y. Chiu, Recursive optimal pruning with applications to tree structured vector quantizers, IEEE Transactions on Image Processing 1 (2) (1992) 162–169.
[16] W.-M. Lam, A.R. Reibman, Quantizer design for decentralized estimation systems with communications constraints, in: Proceedings of the 23rd Annual Conference on Information Sciences and Systems, Baltimore, MD, USA, March 1989, pp. 489–494.
[17] W.-M. Lam, A.R. Reibman, Design of quantizers for decentralized estimation systems, IEEE Transactions on Communications 41 (11) (1993) 1602–1605.
[18] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Transactions on Communications COM-28 (1) (1980) 84–95.
[19] S.P. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory IT-28 (2) (1982) 129–136.
[20] M. Longo, T.D. Lookabaugh, R.M. Gray, Quantization for decentralized hypothesis testing under communication constraints, IEEE Transactions on Information Theory IT-36 (2) (1990) 241–255.
[21] V. Megalooikonomou, Y. Yesha, Quantization for distributed estimation with unknown observation statistics, in: Proceedings of the 31th Annual Conference on Information Sciences and Systems, Baltimore, MD, USA, March 1997.
[22] V. Megalooikonomou, Y. Yesha, Design of neural network quantizers for a distributed estimation system with communication constraints, in: Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing, May 1998.
[23] V. Megalooikonomou, Y. Yesha, Quantizer design for distributed estimation with communication constraints and unknown observation statistics, IEEE Transactions on Communications 48 (2) (2000) 181–184.
[24] A.R. Reibman, L.W. Nolte, Optimal detection and performance of distributed sensor systems, IEEE Transactions on Aerospace and Electronic Systems AES-23 (1) (1987) 24–30.
[25] R.R. Tenney, J. Nils, R. Sandell, Detection with distributed sensors, IEEE Transactions on Aerospace and Electronic Systems AES-17 (4) (1981) 501–510.
[26] H.L. Van Trees, Detection, Estimation, Modulation Theory, Part I, John Wiley and Sons, New York, USA, 1968.
[27] P.K. Varshney, Multisensor data fusion, Electronics and Communication Engineering Journal 9 (6) (1997) 245–253.
[28] B. Warneke, M. Last, B. Liebowitz, K. Pister, Smart dust:communicating with a cubic-millimeter computer, IEEE Computer Magazine (2001) 44–51.
[29] Z. Zhang, T. Berger, Estimation via compressed information, IEEE Transactions on Information Theory IT-34 (2) (1988) 198–211.