MULTI-AGENT LEARNING FRAMEWORK FOR ENVIRONMENT REDUNDANCY IDENTIFICATION FOR MOBILE SENSORS IN AN IOT CONTEXT

D. A. Martínez[1,*], E. Mojica-Nava[1], K. Watson[2], T. Usländer.[2]

[1] National University of Colombia, Electrical and Electronics Engineering Faculty, 111321 Bogotá, Colombia
(damartinezva, eamojican)@unal.edu.co
[2] Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB, 76131 Karlsruhe, Germany
(kym.watson, thomas.uslaender)@iosb.fraunhofer.de

KEY WORDS: Mutual information, Gaussian Process Regression (GPR), multi-agent system, mobile sensor networks, IoT, sensor planning.

ABSTRACT:

From an IoT point of view, the continuous growth of cheap and versatile sensor technologies has generated a massive data flow in communication networks, which most of the time carries unnecessary or redundant information that requires larger storage centers and more time to process and analyze data. Most of this redundancy is due to fact that network nodes are unable to identify environmental cues showing measurement changes to be considered and instead remain at a static location getting the same data. In this work we propose a multi-agent learning framework based on two theoretical tools. Firstly, we use Gaussian Process Regression (GPR) to make each node capable of getting information from the environment based on its current measurement and the measurements taken by its neighbors. Secondly, we use the rate distortion function to define a boundary where the information coming from the environment is neither redundant nor misunderstood. Finally, we show how the framework is applied in a mobile sensor network in which sensors decide to be more or less exploratory by means of the parameter s of the Blahut-Arimoto algorithm, and how it affects the measurement coverage in a spatial area being sensed.

1. INTRODUCTION

The amount of data collected in IoT applications grows continuously due to the high variety of sensing devices and the increase of applications using them. New devices to measure a wide range of variables in surveillance contexts such as air pollution (CO, NO_2, $PM_{2.5}$ and PM_{10}) (Zheng et al., 2013), meteorological data (temperature, wind speed, humidity), assistive living (heartbeat, SpO_2) (González-Valenzuela et al., 2011), disaster management and intelligent transportation are just some examples of the extensive spectrum of applications.

Most of these applications use analytic and statistical methods such as Reinforcement Learning (Song et al., 2013, Sendra et al., 2014), Deep Recurrent Neural Networks (Ma et al., 2015, Ong et al., 2014), Markov Decision Process and Gaussian Process Regression (Cheng et al., 2014, Dongbing Gu and Huosheng Hu, 2012, Kunz et al., 2009) to infer the environment behavior in uncovered zones and to establish collaborative decision making between the network nodes. However, in spite of having good performance in spatial and temporal predictions, these methods are not specifically concerned about the redundant information taken from the environment and its injection through the network from the measurement points, which generates for some applications, unnecessary data accumulation in data centers or repetitive transmission to sink nodes.

On the other hand, from an IoT point of view, applications involving node mobility might have isolated nodes or clusters in a given period of time, which requires network implementations with characteristics such as self-organization and event-driven decision making, so that nodes have abilities to identify environment changes based just in their neighborhood information, and make decisions using them in order to improve features such as transmission times, sleep-awake mode schemes and mobility.

In this regard, in this work we propose a multi-agent framework that allows the nodes (henceforth referred as agents) of a sensor network to identify cues from the environment through inferred information based on its neighborhood knowledge with low information redundancy. The proposed framework consists of the following steps:

- Inference of the environment information for each agent using Gaussian process regression, based on its neighborhood information.
- A decision making scheme that depends on the mutual information of each agent about its environment. To calculate the mutual information between the agent and its environment we use the rate distortion function by means of the Blahut-Arimoto algorithm.
- Agents can change their behavior depending on the input parameter s of the Blahut-Arimoto algorithm. In this way, they can decide where to move in a spatial field depending on the redundancy level of information, becoming more exploratory if they move to positions with low mutual information or less exploratory if they move to positions with high mutual information. This will be shown in section 5.

The rest of this paper is organized as follows. Section 3 describes the rate distortion function and how it can be obtained using the Blahut-Arimoto algorithm. In section 3.2.1 we describe the relevance of the parameter s to define distortion-information points.

[*] Correspondence to: David Martínez, Department of Electrical and Electronics Engineering, National University of Colombia, Bogotá-Colombia: daamartinezva@unal.edu.co

https://doi.org/10.5194/isprs-archives-XLII-4-W11-33-2018

In section 3.3 we show general concepts about Gaussian process regression and its focus in our approach, and in 3.4 we show some graph theory concepts that will be useful to describe the agent interactions when they become more or less exploratory. Finally, in section 4 we describe the proposed framework in a detailed way.

2. RELATED WORK AND CONTRIBUTION

In terms of mobile sensor networks over an IoT context, a huge variety of approaches has been proposed in order to reduce redundancy and energy consumption of sensing devices by means of data collection and data aggregation schemes as is reported in (Ang and Phooi Seng, 2016). In terms of data collection, in (Li et al., 2013), the compressed sensing theory (CS) is used to reduce the sampling points taken for each sensor, achieving therefore, redundant data and energy consumption reduction, since the low sampled data is reconstructed at the base station where the energy is not constrained. In (Wang et al., 2014), the energy consumption is reduced through the scheduling of sleep and awake modes of nodes located in zones where the measures are considered redundant. This approach uses complex networks concepts, specifically a random network model to define through parameters such as the degree distribution, clustering coefficient and shortest path, the minimum number of nodes to maintain the network connected within an specific zone. From the data aggregation point of view, even though most of the approaches mainly aim to increase the network lifetime and reduce the energy consumption, the redundancy reduction is also an issue that researches take into account. In (Zeydan et al., 2012), a game theory adaptive and distributed routing algorithm is proposed. In this approach, the correlated data between nodes is compressed in data streams to reduce the network load. This aggregated data plus the energy and interference define the cost function used by the routing algorithm. In (Scaglione and Servetto, 2005), they generate an estimation on each node about the entire network measurements in order to identify correlated information among measurement points, and in this way, avoid its transmission to a sink through a multi-hop path. This approach uses the rate distortion function so that any node can obtain information from all its peers, under a prescribed distortion condition. These works, in spite of being concerned about the redundant information, are not focused on a multi-agent context where agents can make decisions according to their environment perception. Additionally, some of them require a full information network scheme, i.e, the nodes must be connected with all their peers.

On the other hand, in terms of research combining GPR models (also known as Kriging filter) and multi-agent learning contexts, most of the work has been focused in algorithms to find informative locations through the uncertainty maximization. In (Dongbing Gu and Huosheng Hu, 2012), a distributed Gaussian process regression (DPGR) that only requires neighborhood information is implemented. In this work, they use the information entropy to detect locations with high uncertainties to define an utility function for the central Voronoi tessellation (CVT) algorithm (Cortes et al., 2004). In (Nguyen et al., 2016), a sampling strategy based on entropy maximization is designed to find the most informative locations for mobile robotic wireless sensor networks (MRWSs) over a spatial field modeled with a Gaussian process. In (Oh et al., 2010), they first use sparse Gaussian processes to reduce the computational cost on each sensor before the environment state is predicted. Then, to define an exploratory rule towards the more informative locations, they use and compare three informative strategies: informative vector machine (IVM), principal feature analysis (PFA), and a mutual information based measurement selection algorithm (MI). In (Xu and Choi, 2011) a self-organizing scheme is presented where the agents move in an anisotropic field modeled with a spatio-temporal Gaussian process. A centralized sampling strategy to determine the next position of the agent is proposed based on the minimization of the Fisher Information Matrix. In (Binney et al., 2010), a path planning for underwater vehicles (AUVs) is presented. The movement is based on the minimization of the uncertainty (entropy) between sampled and un-sampled positions. In order to calculate the entropy, they model a underlying scalar field using a Gaussian process. Even though these works are concerned about high or low uncertainty in spatial fields, their main focus is not the redundancy of information and its identification from the agent side.

With this in mind, the main contribution of this work, is the ability of the agents to identify high or low correlated cues from the environment based on the parameter s of the Blahut-Arimoto algorithm, which, *to the best of* our knowledge, has not been discussed in literature. In this sense, agents in an IoT application can decide to be more or less exploratory in an spatial field, always under the limits defined by the rate distortion function, which guaranties a minimum mutual information scheme between agents and their environment, i.e, the minimum redundancy.

3. PRELIMINARIES

3.1 Rate Distortion Theory

In Fig. 1 we show the interaction of an agent and its environment to emulate the emitter-transmitter relationship through a communication channel described in (Shannon, 1948). Here, we represent the information that the agent has about the environment by means of the random variable Y, which can take any value inside of an alphabet $\mathcal{Y}$, having a probability distribution $p(y) = Pr\{Y = y\},\ y \in \mathcal{Y}$. The random variable X, represents the environmental information, with alphabet $\mathcal{X}$ and probability distribution $p(x) = Pr\{X = x\},\ x \in \mathcal{X}$. The difference between X and Y can be calculated using the squared error measure

$$\mathcal{L}(x, y) = (y - x)^2. \tag{1}$$

It means that if the agent has complete certainty about the environmental information, $X = Y$ and $\mathcal{L}(x, y) = 0$. The relationship between X and Y is given by the conditional probability $p(y|x)$, which expresses how the knowledge of Y reduces the uncertainty about X. This can be defined in terms of the entropy as

$$I(X; Y) = H(Y) - H(Y|X = x), \tag{2}$$

where $I(X; Y)$ is the mutual information between X and Y,

$$H(Y) = -\sum_{y\in\mathcal{Y}} p(y) \log p(y) \tag{3}$$

is the entropy of Y, and

$$H(Y|X = x) = -\sum_{x\in\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} p(y|x) \log p(y|x) \tag{4}$$

is the conditional entropy of Y given $X = x$.

It means that the mutual information can take the form

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-4-W11-33-2018 | © Authors 2018. CC BY 4.0 License.

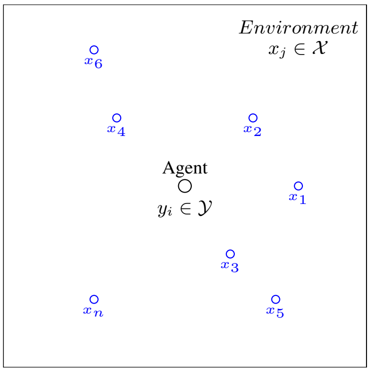


Fig. 1. Agent-environment interaction from a rate distortion theory perspective.

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(y|x)p(x) \log \frac{p(y|x)}{p(y)}. \tag{5}$$

Now, suppose that we want to define the minimum mutual information between X and Y necessary for the agent to understand the environmental information. Logically, this minimization must be subject to an expected distortion value, i.e, for a high mutual information, we must have a low distortion and conversely. In this sense, we have the mutual information minimization problem, also called rate distortion function $R(D)$ given by

$$\begin{aligned} R(D) = \quad & \underset{p(y|x)}{\text{minimize}} I(X;Y) = \sum_{y,x} p(y|x)p(x) \log \frac{p(y|x)}{p(y)} \\ & \text{subject to} \sum_{y,x} p(y|x)p(x)\mathcal{L}(x,y) \le D \\ & \qquad \sum_{y} p(y|x) = 1 \\ & \qquad p(y|x) \ge 0, \end{aligned} \tag{6}$$

where D is the expected distortion value.

Normally, to minimize the mutual information following an analytic procedure is not straightforward. In this regard, the Blahut-Arimoto algorithm offers a less complex procedure to calculate it.

3.2 The Blahut-Arimoto Algorithm

Before explaining the Blahut-Arimoto algorithm, we express (6) through the Lagrange multipliers s[1] and λ. Then

$$\begin{aligned} R(D) = \underset{p(y|x)}{\text{minimize}} \Bigg[& \sum_{y,x} p(y|x)p(x) \log \frac{p(y|x)}{p(y)} \\ & - s \left(\sum_{y,x} p(y|x)p(x)\mathcal{L}(y,x) - D \right) \\ & + \sum_{x} \lambda_x \left(\sum_{y} p(y|x) - 1 \right) \Bigg], \end{aligned} \tag{7}$$

where the inequality restriction $p(y|x) \ge 0$ is temporarily ignored.

Now, let

$$\begin{aligned} J\left[p(y|x), p(y)\right] = & \sum_{y,x} p(y|x)p(x) \log \frac{p(y|x)}{p(y)} \\ & - s \sum_{y,x} p(y|x)p(x)\mathcal{L}(y,x) + \sum_{x} \lambda_x \sum_{y} p(y|x). \end{aligned} \tag{8}$$

Then

$$R(D) = sD - \lambda + \underset{p(y|x)}{\min}\, \underset{p(y)}{\min} J\left[p(y|x), p(y)\right], \tag{9}$$

which is a double minimization problem that is solved in two steps described in appendix B. The results of these two steps, are the basis for the Blahut-Arimoto algorithm definition (Blahut, 1972), which calculates $p^*(y|x)$ and $p^*(y)$ iteratively until a convergence condition is fulfilled (see Algorithm. 1).

```
Algorithm 1: BLAHUT-ARIMOTO
Input: $s, p_0(y)$
Output: $p(y|x), R, D$
$p(y) \leftarrow p_0(y)$
for $i \leftarrow 1$ to $n$ do
    for $j \leftarrow 1$ to $m$ do
        $A_{yx}(i,j) \leftarrow e^{s(x_i - y_j)^2}$
    end
end
while $TU - TL < \epsilon$ do
    $c_y = \sum_x p(x) \frac{A_{yx}}{\sum_y p(y) A_{yx}}$
    $p(y) \leftarrow p(y) c_y$
    $T_U = -\sum_y p(y) \log c_y$
    $T_L = -max \log c_y$
end
$p(y|x) = \frac{A_{xy}}{\sum_y p(y) A_{yx}}$
$D = \sum_y \sum_x p(y)p(y|x)\mathcal{L}(y,x)$
$R = sD - \sum_x p(x) \log \sum_y A_{yx} p(y) - \sum_y p(y) c_y \log c_y$
return $p(y|x), R, D$
```

The choice of the inputs s and $p_o(y)$ follows the next rules:

- $p_o(y) = \frac{1}{|\mathcal{Y}|}$, where $|\mathcal{Y}|$ is the number of elements in $\mathcal{Y}$.
- $s \in \mathbb{R}^-$.

Note that the algorithm requires the prior distribution $p(x)$, which for our approach, is determined by the environment information gathered for each agent. We will describe how to define it in section 4.

3.2.1 The Parameter *s* In order to understand the relevance of parameter s, let's consider a random variable X with $\mathcal{X} = \{x \in \mathbb{R} : -10 \le x \le 10\}$. Assuming that $p(x)$ is a normal distribution given by $\mathcal{N}\left(\mu = 0,\ \sigma^2 = 3\right)$ and applying the Blahut-Arimoto algorithm for $-2 \le s \le -20$, we obtain the curve of the $R(D)$ function shown in Fig. 2(Blahut, 1972). The allowed

[1] We use the symbol s as one of the Lagrange multipliers to be consistent with literature related to the Blahut-Arimoto algorithm.

region depicted in this figure, is composed of the set of all pairs $\{R, D\}$ that guarantee the environmental information recovering for the agent. However, some points in this region might use more information than necessary to represent some data, causing redundancy. The $\{R, D\}$ pairs above the curve $R(D)$, set the limit where the the mutual information is minimum and enough to understand what is happening in the environment. In other words, above the curve we have points where the information is neither redundant nor misunderstood.

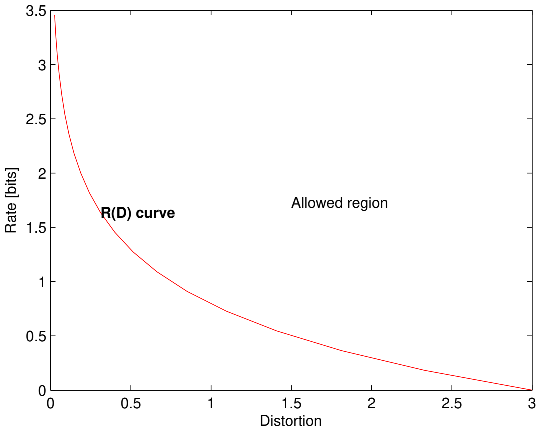


Fig. 2. R(D) curve for $-2 \le s \le -20$ and $p(x) = \mathcal{N}\left(\mu = 0,\ \sigma^2 = 3\right)$

The relationship between the parameter s and the $R(D)$ curve is visible if we recall Eq. (9), which can be rewritten as

$$R(D) = sD + b, \tag{10}$$

i.e, in the form of a linear equation with slope s and intercept

$$b = \min_{p(y|x)p(y)} \min J\left[p(y|x), p(y)\right] - \lambda \tag{11}$$

Then, the value of s determines the slope in a $\{D, R\}$ point over the curve $R(D)$, being the lowest value of s associated to the highest rate and the lowest distortion, as it is shown in Fig. 3.

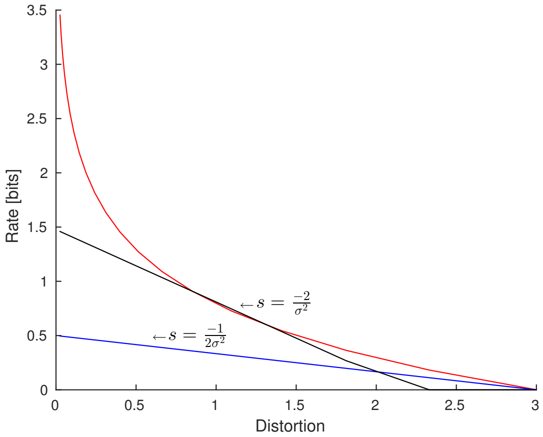


Fig. 3. Relationship between parameter s and $R(D)$ curve.

The parameter s also affects the certainty of the conditional distribution $p(y|x)$. Figure 4 shows how the variance of $p(y|x = -5.15)$ changes depending on the value of s. Note how the variance increases with s.

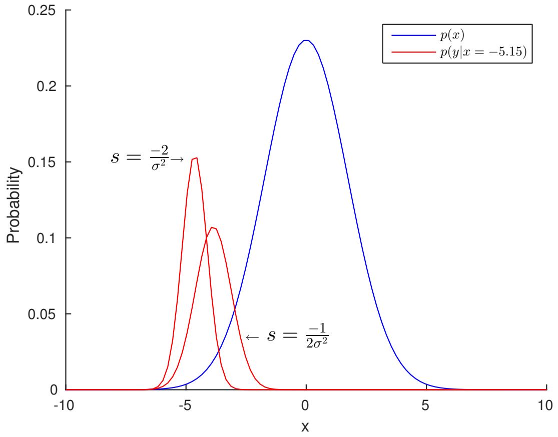


Fig. 4. Relationship between parameter s and $p(y|x)$

At this point is noticeable that the selection of s can define the level of distortion that we want to have for a specific application. In our approach, this parameter is important to establish a *greedy* behavior of agents exploring the environment in a multi-agent system. It will be described in more detail in section 4.

3.3 Non Parametric Learning

In a conventional learning process, the use of linear parametric regressions with the form

$$y = f(x) = w_0 + w_1 x + \epsilon,\ (with\ \epsilon =\ irreducible\, error), \tag{12}$$

has been widely used to predict the behavior of a variable y from a set of known values of x, where the parameters w_n are calculated from an error condition minimization. For the case of two parameters, this minimization is given by

$$\underset{w_o, w_1}{\text{argmin}} \frac{1}{N} \sum_{n=1}^{N} y - f(x_n; w_0, w_1). \tag{13}$$

This kind of regression, in spite of being extended to non linear cases, becomes unmanageable as the number of parameters increases.

In this regard, the GPR offers a non parametric alternative to address regression problems, since it finds a distribution over all possible vectors $f(x)$, coherent with the previously known data (named training data set). A brief explanation of GPR is given for the reader to better understand how it works and is used in our approach.

3.3.1 GPR The GPR consists by the definition of a posterior vector $\boldsymbol{z_k}$, from a prior vector $\boldsymbol{z_h}$ satisfying a *training data set*. The pair (c_{hi}, z_{hi}) denotes the *training point* i in which c_{hi} is a 2-D coordinate and z_{hi} its corresponding value[2]. In the same way, the pair (c_{kj}, z_{kj}) denotes the *testing point* j where c_{kj} is a

[2] Although we are using a 2-D representation for training and test sets, it can be extended to a n-dimensional system.

2-D coordinate and z_{kj} the value that we want to predict. It can be represented as

$$\boldsymbol{c_h} = \begin{bmatrix} c_{h1} \\ \vdots \\ c_{hn} \end{bmatrix}, \boldsymbol{z_h} = \begin{bmatrix} z_{h1} \\ \vdots \\ z_{hn} \end{bmatrix}, \boldsymbol{c_k} = \begin{bmatrix} c_{k1} \\ \vdots \\ c_{km} \end{bmatrix}, \boldsymbol{z_k} = \begin{bmatrix} z_{k1} \\ \vdots \\ z_{km} \end{bmatrix}. \tag{14}$$

Since vectors $\boldsymbol{z_h}$ and $\boldsymbol{z_k}$ are assumed to be Gaussian distributed, their combination must also follow a Gaussian density given by

$$\begin{pmatrix} \boldsymbol{z_h} \\ \boldsymbol{z_k} \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu^* \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix} \right), \tag{15}$$

where K is the training data covariance matrix, K_* the cross-covariance matrix between training and test data, and K_{**} is the covariance matrix for the testing points(Rogers and Girolami, 2016). The calculation of the above covariance matrices, for the case of our approach, is based on the popular kernel RBF (Radial Basis Function), which is defined as

$$c(b_n, b_m) = \alpha e^{-\gamma (b_n - b_m)^2}, \tag{16}$$

where $c(b_n, b_m)$ is the covariance between a pair of points b_n and b_m belonging to any set being evaluated (training or test) and the parameters α and γ define the amplitude and smoothness of the resultant distribution .

Finally, the conditional distribution between the posterior and prior vectors is given by

$$p(\boldsymbol{z_k} | \boldsymbol{z_h}) = \mathcal{N}(\mu^*, \Sigma^*), \tag{17}$$

where

$$\mu^* = K_* K^{-1} \boldsymbol{z_h}, \tag{18}$$

and

$$\Sigma^* = K_{**} - K_*^T K^{-1} K_*. \tag{19}$$

For the calculation of $\boldsymbol{z_k}$ we can use the approximation

$$\boldsymbol{z_k} \approx \mu^* + L\mathcal{N}(0, I), \tag{20}$$

where L is the square root of the covariance matrix, such that $\Sigma^* = LL^T$. The matrix L can be obtained using the Cholesky decomposition.

In the case of our study, the set of predicted values given by $\boldsymbol{z_k}$ represents the environmental information for each agent, which is calculated using the neighborhod information as training data. We will show it in detail in section 5.

3.4 Graph Theory in Multi-Agent Systems

Graph theory and complex networks have high relevance in multi-agent systems description through parameters such as cluster coefficient, average degree and degree distribution, among others. In this section, we explain some concepts related to undirected graphs that have relevance for the results analysis in section 5.

Consider the definition of a graph as $G = (\mathcal{V}, \mathcal{E})$, where the sets $\mathcal{V}$ and $\mathcal{E}$ represent the nodes and the edges connecting them, respectively. An edge denoted as (i, j) links the nodes i and j. If a pair of nodes are connected through an edge, they are said to be *adjacent* or *neighbors*.

A general form to describe the node interaction in a network is by means of the $N \times N$ symmetric *adjacency matrix* $A = \{a_{ij}\}$, which is defined such that

$$a_{ij} = \begin{cases} 1 \; if \; (i,j) \in \mathcal{E} \\ 0 \; if \; (i,j) \notin \mathcal{E} \end{cases}, \tag{21}$$

where N is the number of nodes in the network.

The *degree* of a node i, denoted as k_i is defined as the number of edges connected to it. In terms of the adjacency matrix, it is given by

$$k_i = \sum_j a_{ji}, \tag{22}$$

therefore, the network *average degree* is

$$\langle k \rangle = \frac{1}{N} \sum_i k_i. \tag{23}$$

As we will see in section 5, the value of $\langle k \rangle$ shows how the number of connections of a node changes with the value of the parameter s.

4. AGENT SELF-IDENTIFICATION OF REDUNDANT INFORMATION

So far, we have shown how a GPR can be used to get environment information and predict its behavior from a prior set of *training data*. Additionally, we have explained the usefulness of the $R(D)$ function to get the minimal mutual information between an agent and its environment. Now, let us suppose that we have a multi-agent system in which its individuals get information from the environment just based on the information that they can collect from their neighborhood in order to make decisions. In addition, this collected information is the minimum necessary to understand what is happening in the agent vicinity, avoiding in this way, the use of resources to gather and interpret unnecessary data.

In this regard, we propose the multi-agent learning framework shown in Fig. 5, which combines the environmental-predictive capacity of GPR and the low redundancy characteristics offered by the rate distortion function. This figure shows the steps of the framework, composed of the learning stage, followed by a step where each agent can fix its *greedy* behavior through the parameter s, and the mutual information minimization stage. Firstly, each agent calculates the μ^*, Σ^* representing the covariance and mean values of the predicted set, having as training set the data taken from its neighbors, given by c_h and z_h. Secondly, the parameter s can be tuned for each agent in order to perceive more or less distortion about its environment, allowing it to identify more or less correlated cues and make decisions about them. Thirdly, the environment predicted information, given by μ^*, Σ^*, sets the input probability distribution $p(x)$ of the Blahut-Arimoto algorithm, which is used to calculate $p(z_{ho}|z_k)$, i.e, the conditional probability between the agent information z_{ho} and the environment information z_k. Finally, the agent modifies its behavior according to a strategy β, which indicates the level of mutual information to follow, i.e, if it will be more of less exploratory.

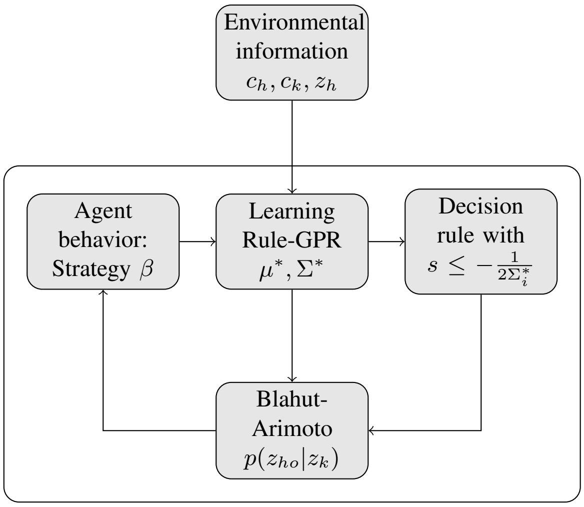


Fig. 5. Proposed multi-agent learning framework.

4.1 The *Greedy* behavior of Parameter s

As we mentioned before, the value of the parameter s determines the pair $\{R, D\}$ on the $R(D)$ curve. In our model, the choice of this parameter for each agent depends on the calculated variance Σ_j^* in the predicted point c_{kj}. It can be understood if we consider the normal distribution $\mathcal{N}(\mu_j^*, \Sigma_j^*)$ as the input probability distribution $p(x)$ of the Blahut-Arimoto algorithm, which, according to (Cover and Thomas, 2005), lets us to calculate $R(D)$ through the expression

$$R(D) = \begin{cases} \frac{1}{2}\log\frac{\Sigma_j^*}{D}, & 0 \leq D \leq \Sigma_j^* \\ 0, & D > \Sigma_j^*. \end{cases} \tag{24}$$

Thus, considering that s is the slope of curve $R(D)$ in a point $\{D, R\}$, we have

$$s = \frac{dR(D)}{dD} = -\frac{1}{2D}, \tag{25}$$

which, evaluated in the maximum distortion value, gives us the upper boundary condition

$$s \leq -\frac{1}{2\Sigma_j^*}. \tag{26}$$

For this value of s, the distortion on each agent about the environmental information is the highest. However, we still are inside of the values allowed by the rate distortion function. In this point, agents demonstrate to be more exploratory (*greedy*) than for values of s associated to lower distortions, where they prefer to follow the same environmental cues followed by their peers.

In section 5, we will show how the proposed model behaves in a mobile sensor network simulated inside of a 2-D grid where a measured variable has different levels on each point. The results will show the network behavior for different values of the parameter s, generating more or less exploratory interest on each agent (sensor).

4.2 Algorithm Representation

The Algorithm 2 shows a brief representation of the framework implementation. Observe how the Blahut-Arimoto algorithm is included to calculate the conditional probability between the agent measurement z_{h0} and the environment predictions represented as z_{kj}. The returned parameter β describes the strategy to be followed by the agent, which in the case of a mobile sensor network, as we will see in section 5, is given by the movement towards the estimated point c_{kj} with the lowest mutual information.

Algorithm 2: Self-Redundancy Identification

```
Input: $c_h, z_h, c_k$
Output: $\beta$
$n \leftarrow length(c_k)$
$\mu^* \leftarrow KK^{-1}z_h$
$\Sigma^* \leftarrow K_* - K_*^T K^{-1} K_*$
for $j \leftarrow 1$ to $n$ do
    $p(z_{kj}) \leftarrow \mathcal{N}(\mu_j^*, \Sigma_j^*)$
    $s \leq -\frac{1}{2\Sigma_j^*}$
    $p(z_{ho}|z_{kj}) \leftarrow$ BLAHUT-ARIMOTO$(p(z_{kj}), s)$
end
$\beta \leftarrow min\, p(z_{ho}|z_k)$
return $\beta$
```

5. CASE OF STUDY: A MOBILE SENSOR NETWORK

The framework implementation is carried out in a mobile sensor network composed of 10 sensors (agents) with mobility capabilities deployed randomly in a 2-D grid that simulates a spatial field (see Fig. 6). Observe the variation of the hypothetical measure along the 2D-grid and the initial positions of the agents.

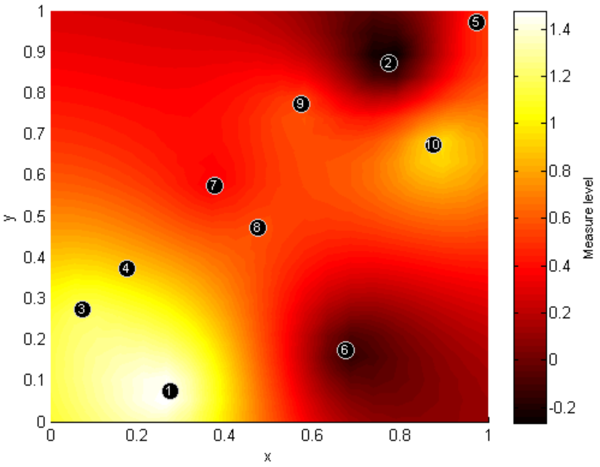


Fig. 6. Initial grid with 10 mobile agents.

The ***agent neighborhood definition*** is shown in Fig. 7. Here we can see two circular regions surrounding a set of agents (filled black dots) and some predictive points (red circles). The inner circle corresponds to the coverage range of the agent located in its center. Agents inside this circle form a neighborhood and have a connection with the central one. The external circle contains the prediction points with known coordinates c_{kj} and predicted values z_{kj}. The prediction is made for the agent using as training data the values z_{hi} at coordinates c_{hi} corresponding to the measures and positions of the neighborhood agents. Once the prediction is done, the agent decides to move in the direction established by the minimal mutual information, looking for the most dissimilar location each time.

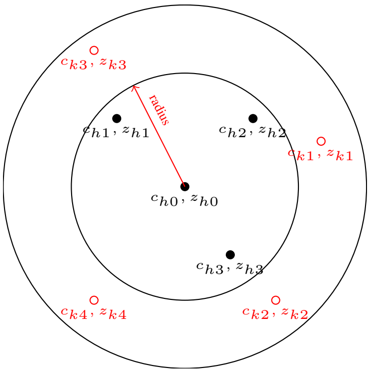


Fig. 7. Agent neighborhood.

5.1 The Greedy Effect of s

The exploratory effect of parameter s on each agent is shown in Fig. 8. In Fig. 8a, $s = -10$. In this case, near agents exhibit preference for the same environment cue, following similar trajectories and generating clusters; therefore, the coverage is reduced. In Fig. 8b, we use the higher boundary $s = \frac{-1}{2\Sigma_j^*}$. In this case, the agents become more exploratory, improving the coverage.

In Fig. 9 the trajectories for all agents are shown for 20 simulation steps and different values of s. In Fig. 9a, $s = -10$. In this case, agents exhibit redundant trajectories, which results in loss of coverage in some regions. In Fig. 9b, $s = \frac{-4}{\Sigma_j^*}$. The coverage in this case improves but some agents still explore regions with redundant measures. In Fig. 9c, $s = \frac{-1}{2\Sigma_j^*}$, i.e, the high boundary for the parameter s. In this case, agents show a preference to move through regions where the environment is more variable. In terms of coverage, it has a better performance than the two previous cases, since the exploration is reduced in zones where the measures are redundant.

Results in Fig. 10 show the variation of the degree for each one of the 10 agents when the value of s changes. Additionally, it is possible to note how the average degree of the system increases as the value of s decreases beyond $\frac{-4}{\Sigma_j^*}$. Agents following cues with low distortion (lower values of s) tend to form clusters and increase their average degree. In terms of data transmission, it means that redundant information is collected and sent to the data center or sink. On the other hand, agents following distorted cues (higher values of s), tend to have lower average degrees, avoiding cluster formation and redundancy in data centers.

In spite of the apparent differences in the agents behavior for different values of s, it is important to note that any selected value below the upper boundary condition is a point in the rate distortion curve, and therefore, is a point for minimal mutual information, i.e, the agents make decisions using just the necessary information from the environment.

6. CONCLUSIONS

Our study provides a framework for learning in multi-agent systems based on GPR to gather predictive environment information and the rate distortion function to define conditions for agent decision making under low redundancy conditions. The results, implemented in a mobile sensor network, shown how the parameter

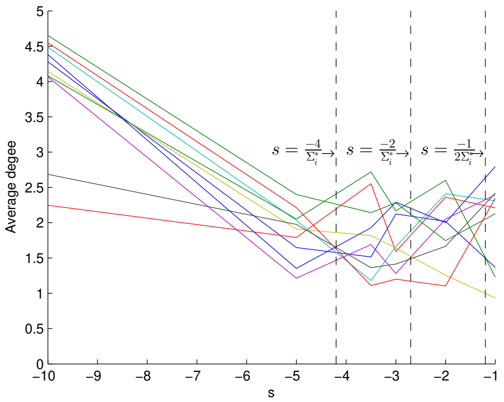


Fig. 10. Average degree for different values of s.

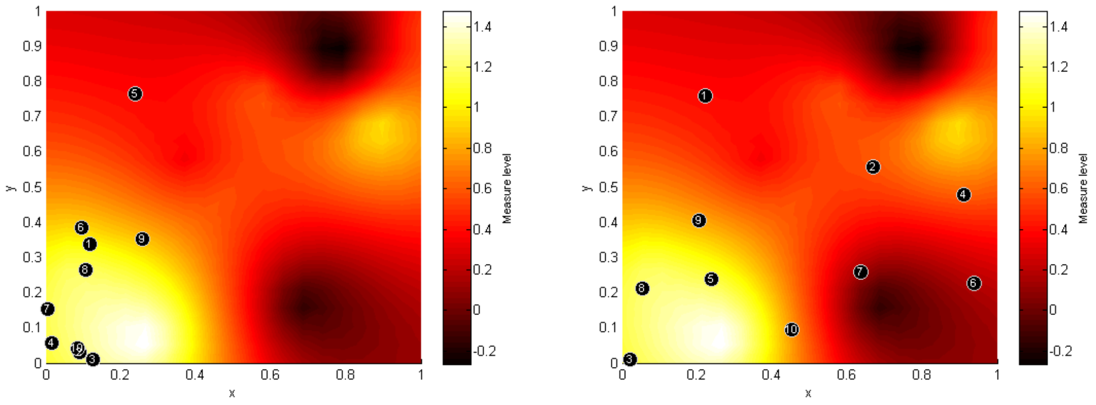
s of the Blahut-Arimoto algorithm allows the agents to be more or less exploratory. This behavior can improve coverage efficiency and reduce the transmission of redundant data.

On the other hand, we have shown how the network average degree maintains lower values when s is closer to its upper boundary. It means that the agents maintain a low number of connections because they are continuously moving to uncovered regions. Additionally, agents show a preference to move in zones where the environment is variable, avoiding the exploration of places having similar measurements.

Since the agents have the ability to detect locations with redundant information, an energy waste reduction is implicit when they avoid to move in those directions. Additionally, agents do not require a full information scheme, i.e, they do not need to get information from all their peers in order to make predictions about the environment, which can be traduced in a reduction of the communication cost.

ACKNOWLEDGEMENTS

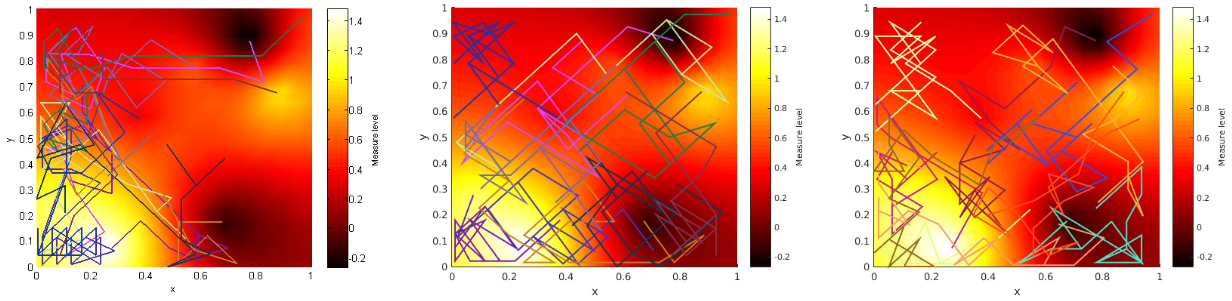
This research has been supported by a scholarship from The Fundación de Ciencia y Tecnología Colombo-Alemana (FunCyTCA) awarded to D. A. Martínez.



(a)

(b)

Fig. 8. The exploratory effect of parameter s on each agent. (a) For $s = -10$. Near agents exhibit preference for the same environment cue. (b) $s = \frac{-1}{2\Sigma_j^*}$. The agents become more exploratory.



(a)

(b)

(c)

Fig. 9. Coverage effect of parameter s. (a) $s = -10$. Agents have redundant trajectories and some regions remain uncovered. (b) $s = \frac{-4}{\Sigma_j^*}$. The coverage improves but agents still explore redundant zones. (c) $s = \frac{-1}{2\Sigma_j^*}$. Agents tend to explore regions where the environment is more variable.

REFERENCES

Ang, L.-M. and Phooi Seng, K., 2016. Big Sensor Data Applications in Urban Environments. *Big Data Research* 4, pp. 1–12.

Binney, J., Krause, A. and Sukhatme, G. S., 2010. Informative path planning for an autonomous underwater vehicle. In: *2010 IEEE International Conference on Robotics and Automation*, IEEE, pp. 4791–4796.

Blahut, R., 1972. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory* 18(4), pp. 460–473.

Cheng, Y., Li, X., Li, Z., Jiang, S. and Jiang, X., 2014. Fine-grained air quality monitoring based on gaussian process regression. In: C. K. Loo, K. S. Yap, K. W. Wong, A. Teoh and K. Huang (eds), *Neural Information Processing*, Springer International Publishing, Cham, pp. 126–134.

Cortes, J., Martinez, S., Karatas, T. and Bullo, F., 2004. Coverage Control for Mobile Sensing Networks. *IEEE Transactions on Robotics and Automation* 20(2), pp. 243–255.

Cover, T. M. and Thomas, J. A., 2005. *Elements of Information Theory*.

Dongbing Gu and Huosheng Hu, 2012. Spatial Gaussian Process Regression With Mobile Sensor Networks. *IEEE Transactions on Neural Networks and Learning Systems* 23(8), pp. 1279–1290.

González-Valenzuela, S., Chen, M. and Leung, V. C. M., 2011. Mobility support for health monitoring at home using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine* 15(4), pp. 539–549.

Kunz, S., Usländer, T. and Watson, K., 2009. A testbed for sensor service networks and the fusion SOS: Towards plug & measure in sensor networks for environmental monitoring with OGC standards. pp. 973–979.

Li, S., Xu, L. D. and Wang, X., 2013. Compressed Sensing Signal and Data Acquisition in Wireless Sensor Networks and Internet of Things. *IEEE Transactions on Industrial Informatics* 9(4), pp. 2177–2186.

Ma, X., Yu, H., Wang, Y. and Wang, Y., 2015. Large-scale transportation network congestion evolution prediction using deep learning theory. In: *PloS one*.

Martinez, D. A., Rincon, R., Mojica-Nava, E. and Pavas, A., 2017. Reactive power sharing in microgrids: An information-theoretical approach. In: *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, IEEE, pp. 1–6.

Nguyen, L. V., Kodagoda, S., Ranasinghe, R. and Dissanayake, G., 2016. Information-Driven Adaptive Sampling Strategy for Mobile Robotic Wireless Sensor Network. *IEEE Transactions on Control Systems Technology* 24(1), pp. 372–379.

Oh, S., Xu, Y. and Choi, J., 2010. Explorative navigation of mobile sensor networks using sparse Gaussian processes. In: *49th IEEE Conference on Decision and Control (CDC)*, IEEE, pp. 3851–3856.

Ong, B. T., Sugiura, K. and Zettsu, K., 2014. Dynamic pre-training of deep recurrent neural networks for predicting environmental monitoring data. In: *2014 IEEE International Conference on Big Data (Big Data)*, pp. 760–765.

Rogers, S. and Girolami, M., 2016. *A First Course in Machine Learning, Second Edition*. 2nd edn, Chapman & Hall/CRC.

Scaglione, A. and Servetto, S., 2005. On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks. *Wireless Networks* 11(1-2), pp. 149–160.

Sendra, S., Granell, E., Lloret, J. and Rodrigues, J. J. P. C., 2014. Smart collaborative mobile system for taking care of disabled and elderly people. *Mobile Networks and Applications* 19(3), pp. 287–302.

Shannon, C. E., 1948. A Mathematical Theory of Communication. *Bell System Technical Journal* 27(3), pp. 379–423.

Song, X., Zhang, Q., Sekimoto, Y., Horanont, T., Ueyama, S. and Shibasaki, R., 2013. Intelligent system for human behavior analysis and reasoning following large-scale disasters. *IEEE Intelligent Systems* 28(4), pp. 35–42.

Wang, H., Roman, H. E., Yuan, L., Huang, Y. and Wang, R., 2014. Connectivity, coverage and power consumption in large-scale wireless sensor networks. *Computer Networks* 75, pp. 212–225.

Xu, Y. and Choi, J., 2011. Adaptive Sampling for Learning Gaussian Processes Using Mobile Sensor Networks. *Sensors* 11(3), pp. 3051–3066.

Zeydan, E., Kivanc, D., Comaniciu, C. and Tureli, U., 2012. Energy-efficient routing for correlated data in wireless sensor networks. *Ad Hoc Networks* 10(6), pp. 962–975.

Zheng, Y., Liu, F. and Hsieh, H.-P., 2013. U-air: When urban air quality inference meets big data. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, ACM, New York, NY, USA, pp. 1436–1444.

APPENDIX A. KULLBACK-LEIBLER DIVERGENCE

The Kullback-Leibler divergence, also known as relative entropy, is a measure of the *distance* between two probability distributions. It is commonly used to find the *gain or loss of information* obtained for describing with a distribution Q a random variable X whose original distribution is P. This difference is calculated as

$$D(P||Q) = \sum P \log \frac{P}{Q} \; [bits], \tag{27}$$

which is always non-negative and equal to zero if and only if $P = Q$.

APPENDIX B. STEPS FOR MUTUAL INFORMATION MINIMIZATION

The double minimization problem to solve equation (9) follows the next two steps:

1. For fixed $p(y)$: Due to $I(y, x)$ is a convex function for $p(y|x)$, we have

$$\begin{aligned} R(D) &= \frac{\partial}{\partial p(y|x)} \Big[sD - \lambda + J\left[p(y|x), p(y)\right] \Big] \\ &= p(x) \log p(y|x) + p(x) - p(x) \log p(y) \\ &- sp(x)\mathcal{L}(y, x) + \lambda = 0. \end{aligned} \tag{28}$$

If we make $\log u = 1 + \frac{\lambda}{p(x)}$, then

$$p(y|x) = \frac{p(y)e^{s\mathcal{L}(y,x)}}{u}. \tag{29}$$

Since $\sum_y p(y|x) = 1$,

$$u = \sum_y p(y)e^{s\mathcal{L}(y,x)}, \tag{30}$$

i.e, λ is selected to accomplish the condition $\sum_y p(y|x) = 1$. Therefore, the initially ignored condition $p(y|x) \geq 0$ is also satisfied.

Hence,

$$p^*(y|x) = \frac{p(y)e^{s\mathcal{L}(y,x)}}{\sum_y p(y)e^{s\mathcal{L}(y,x)}}. \tag{31}$$

2. For fixed $p(y|x)$: In this case, it is enough to probe that

$$I(X; Y) = \min_{p(y)} J\left[p(y)\right]. \tag{32}$$

Then

$$\begin{aligned} I(X; Y) - \sum_{y,x} p(y|x)p(x) \log \frac{p(y|x)}{p^*(y)} &= 0 \\ \sum_{y,x} p(y|x)p(x) \log \frac{p(y|x)}{p^(y)} - \sum_{y,x} p(y|x)p(x) \log \frac{p(y|x)}{p^*(y)} &= 0 \\ \sum_{y,x} p(x)p(y|x) \log \frac{p^*(y)}{p(y)} &= 0. \end{aligned} \tag{33}$$

Setting $\sum_x p(x)p(y|x) = p^*(y)$ and using the Kullback-Leibler divergence (see appendix A), we obtain

$$\sum_y p^*(y) \log \frac{p^*(y)}{p(y)} = D(p^*||p) = 0, \tag{34}$$

which means that the equality is satisfied only if the probability distributions p and p^* are the same. Therefore,

$$p^*(y) = \sum_x p(x)p(y|x). \tag{35}$$