Abstract: The success of mobile robots relies on the ability to extract from the environment additional information beyond simple spatial relations. In particular, mobile robots need to have semantic information about the entities in the environment such as the type or the name of places or objects. This work addresses the problem of classifying places (room, corridor or doorway) using mobile robots equipped with a laser range scanner. This paper compares the results of several AdaBoost algorithms (Viola-Jones AdaBoost, Gentle AdaBoost, Modest AdaBoost and Generalized AdaBoost for the place categorization) to train a set of classifiers and discusses these solutions. Since the problem is multi-class and these AdaBoosts provide only binary outputs, the AdaBoosts are arranged into Probabilistic Decision Lists (PDL), where each AdaBoost of the list gives a confidence value of each class. Then, Probabilistic Relaxation Labeling (PRL) is performed to smooth the classification results. Moreover, heuristics for removing incorrect regions are proposed to reduce the classification error. Experimental results suggest that PDL can be extended to several binary classifiers and show that PRL improves significantly the classification rates of the classifiers.

Keywords: intelligence; learning; pattern recognition; mobile robot; semantic place labeling.

I. Introduction

Mobile robots must be able to interact and extract information from their environments. Tasks can be performed by being able to recover useful spatial descriptions of its surroundings using sensors and by utilizing these information in appropriate short-term and long-term planning and decision making activities [1, 2, 3]. Robot-built maps have been used for tasks like path planning, navigation and localization [4]. The large majority of the maps are limited in describing the environment as occupied or unoccupied areas, or with other spatial and/or metric information. However, they neglect descriptions of environment aspects, such as, its navigability or the nature of the activity that typically occurs there. Semantic mapping consists of using mobile robots to built maps that incorporate not only metric occupancy but also other important descriptions of the environment [5]. Semantic information can be used to describe about the functionalities of objects and environments, or to offer additional information to the navigation and localization systems [6]. Additionally, semantic information is crucial to the ability of the robot to interact with humans using some set of terms and concepts [7]. The process of applying a semantic term to some division of the environment is known as semantic place labeling. In this case, semantic information is used to classify places in the environment in categories, such as, room, doorway, corridor and hallway [8].

Besides the quality of sensor information, the success of semantic place labeling relies on the classification system used to categorize places. Some approaches to address this problem have been proposed in the literature. Buschka and Saffiotti [9] propose a local technique uses range data to detect places during navigation. Logistic Regression for indoor scenarios using a multi-class classifier is investigated by Shi and Kodagoda [10]. Moreover, Mozos et al. [11] proposes the Generalized AdaBoost [12] for the place categorization based on the laser range data.

AdaBoost has been a powerful algorithm for solving classification problems. AdaBoost is based on the idea of combining a set of classifiers (weak hypotheses) to form a stronger classifier. AdaBoost was introduced by Freund and Schapire [13] and this algorithm is commonly referenced as Classic AdaBoost. Schapire and Singer [14] proposed some improvements to the Classic AdaBoost, resulting in a new approach called Generalized AdaBoost. The main contributions of the Generalized AdaBoost are the inclusion of confidence-rated predictions for each weak hypothesis and a parameter that controls the influence of each weak hypothesis. Other important contribution for the AdaBoost algorithms is the Viola-Jones AdaBoost, proposed by Viola and Jones [12]. The Viola-Jones AdaBoost ensures that each weak hypothesis depends on only single feature, reducing the number of available features and focusing on a small set of critical features. Since the Viola-Jones AdaBoost uses the Classic AdaBoost’s structure, Mozos et al. [15] proposed a Generalized AdaBoost for the place categorization using the Generalized AdaBoost’s structure and weak hypotheses depending only on a single feature (as the Viola-Jones AdaBoost).

Furthermore, experimental results suggest that Gentle AdaBoost [16] and Modest AdaBoost [17] outperform other classifiers [18], such as, Support Vector Machines [19] and

Semantic Place Labeling Using a Probabilistic Decision List of AdaBoost Classifiers

Symone G. Soares¹ and Rui Araújo²

Institute of Systems and Robotics (ISR-UC), Department of Electrical and Computer Engineering (DEEC-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal
¹symonesoares@isr.uc.pt
²raui@isr.uc.pt

¹Department of Electrical and Computer Engineering (DEEC-UC), University of Coimbra, Pólo II, PT-3030-290 Coimbra, Portugal
Neural Networks [20]. Empirical studies indicate that Gentle AdaBoost has similar performance when compared to other AdaBoosts, however it often outperforms them in terms of stability. On the other hand, the Modest AdaBoost has good generalization capability when compared to the other AdaBoosts.

This paper addresses the problem of classifying places using a mobile robot equipped with a laser range scanner. The environments are categorized as room, corridor or doorway. This work explores the use of the Viola-Jones AdaBoost, the Gentle AdaBoost and the Modest AdaBoost as models and then it compares these models with the Generalized AdaBoost for place categorization. Since the problem is multiclass and these AdaBoosts provide only binary outputs, the AdaBoosts are arranged into the Probabilistic Decision Lists (PDL), where each AdaBoost of the list gives a confidence value of each class. After classification by the PDL, Probabilistic Relaxation Labeling (PRL) is performed to smooth the classifications. The results indicate significant improvements on the Gentle AdaBoost and the Modest AdaBoost when the PRL is applied. Moreover, heuristics for removing incorrect regions are proposed to reduce the error on test data, thus improving generalization. The paper presents experiments that use two data sets which correspond to the Building 52 and the Building 79 at the University of Freiburg [8]. Experimental results suggest that PDL can be extended to several binary classifiers and show that PRL improves significantly the classification rates of the classifiers.

The paper is organized as follows. Section II reports the description of the features acquired from the laser sensors for semantic place labeling. Section III introduces a review of the AdaBoost algorithms. In Section IV, the problem of designing the PDL is addressed. Section V reviews concepts about the PRL. Section VI proposes a set of heuristics for improving the classification rate. Section VII reports experimental results of the AdaBoost using the PDL, the PRL and the heuristics. Finally, Section VIII contains concluding remarks.

## II. Features for the Place Semantic Labeling

This work proposes an approach that allows a mobile robot to build a semantic map from sensor data and then use this information for place labeling. The aim is to classify the robot position based on the current scan obtained from the range sensor. The experiments use two data sets which correspond to Building 52 and Building 79 at the University of Freiburg [8].

It is assumed that the mobile robot is equipped with a laser range scanner that covers a 360° field of view around the robot. Each observation $z = \{b_0, \ldots, b_{M-1}\}$ contains a set of beams $b_i$ and each beam $b_i$ consists of a tuple $(\rho_i, d_i)$, where $\rho_i$ is the angle of the beam relative to the robot and $d_i$ is the length of the beam. The training data set is given by $E = \{(z_i, y_i) \mid y_i \in Y\}$, where $Y = \{\text{room, corridor, doorway}\}$ is the set of possible classes of the places.

The data set is composed of two sets of features that are calculated for each observation. The first set is calculated using the raw beam in $z$. Table 1 shows a list of the single-valued features obtained from raw beams and composing the set $B$. The second set of features, $P$, is obtained from a polygonal approximation $P(z)$ of the area covered by $z$. The points of $P(z)$ correspond to the coordinates of the endpoints of each beam $b_i$ of $z$ relative to the mobile robot: $P(z) = \{(d_i \cos \rho, d_i \sin \rho) \mid \rho = 0, \ldots, M - 1\}$. The list of the features corresponding to the set $P$ is presented in Table 2. This work uses the features extracted from sets $B$ and $P$ [8]. Table 3 lists and defines the features that are used in this paper.

### Table 1: Features for the place categorization using raw beams - Set $B$

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>The average difference between the length of consecutive beams.</td>
</tr>
<tr>
<td>B.2</td>
<td>The standard deviation of the difference between the length of consecutive beams.</td>
</tr>
<tr>
<td>B.3</td>
<td>Same as feature B.1, but considering different max-range values.</td>
</tr>
<tr>
<td>B.4</td>
<td>Same as feature B.2, but considering different max-range values.</td>
</tr>
<tr>
<td>B.5</td>
<td>The average beam length.</td>
</tr>
<tr>
<td>B.6</td>
<td>The standard deviation of the beam length.</td>
</tr>
<tr>
<td>B.7</td>
<td>Number of gaps in the scan. Two consecutive beams build a gap if their difference is greater than a given threshold. Different features are used for different threshold values.</td>
</tr>
<tr>
<td>B.8</td>
<td>Number of beams lying on lines that are extracted from the range scan.</td>
</tr>
<tr>
<td>B.9</td>
<td>Euclidean distance between two points corresponding to two consecutive global minima.</td>
</tr>
<tr>
<td>B.10</td>
<td>Angular distance between two points corresponding to two consecutive global minima.</td>
</tr>
<tr>
<td>B.11</td>
<td>Average of the relation between the length of two consecutive beams.</td>
</tr>
<tr>
<td>B.12</td>
<td>Standard deviation of the relation between the length of two consecutive beams.</td>
</tr>
<tr>
<td>B.13</td>
<td>Average of normalized beam length.</td>
</tr>
<tr>
<td>B.14</td>
<td>Standard deviation of normalized beam length.</td>
</tr>
<tr>
<td>B.15</td>
<td>Number of relative gaps.</td>
</tr>
<tr>
<td>B.16</td>
<td>Kurtosis.</td>
</tr>
</tbody>
</table>

### III. AdaBoost Algorithms

The AdaBoost algorithm has been a very successful technique for solving two-class classification problems. The main idea is to combine a set of weak classification functions to form a stronger classifier, $H$. The classifiers are called weak classification functions because it is not expected that they have the best classification functions to classify the training data well. Weak learner commonly denotes a generic weak learning algorithm. On each round $t$ ($t = 1, \ldots, T$) the weak learner searches over a set of possible weak classification functions and then selects a weak classification function based on the minimization of the classification error. Here, this selected weak classification function is called as weak hypothesis, $h_t$.

After each round, the samples are re-weighted in order to emphasize those which were incorrectly classified by the previous weak hypothesis. The distribution of the weights $D$ indicates the importance of the samples. The final strong clas-
Table 2: Features for the place categorization using polygonal approximation - Set P.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P.1</td>
<td>Area of $P(z)$.</td>
</tr>
<tr>
<td>P.2</td>
<td>Perimeter of $P(z)$.</td>
</tr>
<tr>
<td>P.3</td>
<td>Area of $P(z)$ divided by Perimeter of $P(z)$.</td>
</tr>
<tr>
<td>P.4</td>
<td>Mean distance between the centroid and the shape boundary.</td>
</tr>
<tr>
<td>P.5</td>
<td>Standard deviation of the distances between the centroid and the shape boundary.</td>
</tr>
<tr>
<td>P.6</td>
<td>Similarity invariant descriptors based in the Fourier transformation. It Is used the first 200 descriptors.</td>
</tr>
<tr>
<td>P.7</td>
<td>Major axis $Ma$ of the ellipse that approximates $P(z)$ using the first two Fourier coefficients.</td>
</tr>
<tr>
<td>P.8</td>
<td>Minor axis $Mf$ of the ellipse that approximates $P(z)$ using the first two Fourier coefficients.</td>
</tr>
<tr>
<td>P.9</td>
<td>$Ma/Mf$.</td>
</tr>
<tr>
<td>P.10</td>
<td>Seven invariants calculated from the central moments of $P(z)$.</td>
</tr>
<tr>
<td>P.11</td>
<td>Normalized feature of compactness of $P(z)$.</td>
</tr>
<tr>
<td>P.12</td>
<td>Normalized feature of eccentricity of $P(z)$.</td>
</tr>
<tr>
<td>P.13</td>
<td>Form factor of $P(z)$.</td>
</tr>
<tr>
<td>P.14</td>
<td>Circularity of $P(z)$.</td>
</tr>
<tr>
<td>P.15</td>
<td>Normalized circularity of $P(z)$.</td>
</tr>
<tr>
<td>P.16</td>
<td>Average normalized distance between the centroid and the shape boundary.</td>
</tr>
<tr>
<td>P.17</td>
<td>Standard deviation of the normalized distance between the centroid and the shape boundary.</td>
</tr>
</tbody>
</table>

Table 3: Features used in this work and extracted from the Sets B and P.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>feature B.1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>feature B.2</td>
</tr>
<tr>
<td>$x_3$</td>
<td>feature B.3</td>
</tr>
<tr>
<td>$x_4$</td>
<td>feature P.1</td>
</tr>
<tr>
<td>$x_5$</td>
<td>feature P.2</td>
</tr>
<tr>
<td>$x_6$</td>
<td>feature P.3</td>
</tr>
<tr>
<td>$x_7$</td>
<td>feature B.5</td>
</tr>
<tr>
<td>$x_8$</td>
<td>feature B.6</td>
</tr>
<tr>
<td>$x_9$</td>
<td>feature P.4</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>feature P.5</td>
</tr>
<tr>
<td>$x_{11,\ldots, x_{30}}$</td>
<td>feature B.7 with thresholds 0.5 [m] to 10.0 [m] (0.5 [m] steps)</td>
</tr>
<tr>
<td>$x_{31,\ldots, x_{129}}$</td>
<td>feature P.6 with Fourier coefficients 2 to 100</td>
</tr>
<tr>
<td>$x_{130}$</td>
<td>feature P.7</td>
</tr>
<tr>
<td>$x_{131}$</td>
<td>feature P.8</td>
</tr>
<tr>
<td>$x_{132}$</td>
<td>feature P.9</td>
</tr>
<tr>
<td>$x_{133}$</td>
<td>feature B.11</td>
</tr>
<tr>
<td>$x_{134}$</td>
<td>feature B.12</td>
</tr>
<tr>
<td>$x_{135}$</td>
<td>feature B.13</td>
</tr>
<tr>
<td>$x_{136}$</td>
<td>feature B.14</td>
</tr>
<tr>
<td>$x_{137}$</td>
<td>feature P.16</td>
</tr>
<tr>
<td>$x_{138}$</td>
<td>feature P.17</td>
</tr>
<tr>
<td>$x_{139,\ldots, x_{148}}$</td>
<td>feature B.15 with thresholds 0.1 [m] to 1.0 [m] (0.1 [m] steps)</td>
</tr>
<tr>
<td>$x_{149}$</td>
<td>feature P.15</td>
</tr>
<tr>
<td>$x_{150}$</td>
<td>feature B.16</td>
</tr>
</tbody>
</table>

Algorithm 1 Viola-Jones AdaBoost.

1. Given samples $(x_i, y_i), i = 1, \ldots, N$, where $y_i = 0$ for the negative samples and $y_i = 1$ for the positive samples, respectively. And a set of features $f_j, j = 1, \ldots, M$, so that for each sample $i$ there is a value of $f_j$ given by $f_j(x_i)$;

2. Initialize a distribution as $D_1(i) = \frac{1}{N}$ for $y_i = 0$ and $D_1(i) = \frac{1}{N}$ for $y_i = 1$, where $m$ and $l$ are the number of negative and positive samples, respectively;

3. Set $t \leftarrow 1$;

4. Repeat

(a) Normalize the distribution so that $\sum_{i=1}^{N} D_t(i) = 1$;

(b) For each feature $j$, train a classifier $h_j$ which is restricted to using a single feature. Evaluate the error of each classifier $h_j$ based on the $D_t$: $\varepsilon_t = \sum_{i=1}^{N} D_t(i)[h_j(x_i) - y_i]$;

(c) Choose the classifier $h_t$ with the lowest error $\varepsilon_t$;

(d) Set $\beta_t = \varepsilon_t/(1 - \varepsilon_t)$;

(e) Update the distribution as: $D_{t+1}(i) = D_t(i)\beta_t^{-[h_t(x_i) - y_i]}$;

(f) Set $t \leftarrow t + 1$;

until $t = T$;

5. Construct the final strong classifier as:

$$H(x) = \begin{cases} 1, & \text{if } \sum_{t=1}^{T} (\log \frac{1}{\beta_t})h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \log \frac{1}{\beta_t}, \\ 0, & \text{otherwise.} \end{cases}$$

The AdaBoost was firstly introduced by Freund and Schapire [13] through the Classic AdaBoost algorithm. Afterwards, Schapire and Singer [14] proposed some improvements to Classic AdaBoost, resulting in a new algorithm called Generalized AdaBoost. The main difference is that the Generalized AdaBoost considers that the weak hypothesis can produce output real-valued or confidence-rated predictions. In other words, given an input $x_i$, if $h_t(x_i)$ is close to or far from zero, it is considered as a low or right confidence prediction. Another differing aspect is the inclusion of parameters $\alpha_t$ that control the influence of the weak hypotheses in the final output. The main effect of $\alpha_t$ is to decrease or increase the weights of the training samples classified correctly or incorrectly by the $h_t$.

A. Viola-Jones AdaBoost and Generalized AdaBoost for the Place Categorization

The traditional AdaBoosts are procedures for selecting a small set of good weak hypotheses. However, such weak hypothesis often do not have significant variety. To compensate this problem, Viola and Jones [12] proposed an approach for constructing a classifier by selecting a small number of features using the Classic AdaBoost Algorithm [13]. During the learning procedure a large majority of the available features are excluded, focusing on a small set of critical features. In other works, the weak learner restricts each weak hypothesis to depend on only a single feature. Algorithm 1 details the Viola-Jones AdaBoost. Considering a binary classification problem where the aim is to dis-
criterion for positive and negative samples, the Viola-Jones AdaBoost takes as input a set of samples \((x_1, y_1), \ldots, (x_N, y_N)\), where each sample \(x_i\) belongs to some domain space \(X\) and \(y_i\) belongs to the label set \(Y = \{0, 1\}\). On each iteration \(t\), a distribution \(D_t\) is computed by normalizing the weights.

The weak learner algorithm is employed to select a single feature which best separates the positive and negative samples. In this process, the weak learner finds the optimal threshold classification function so that the minimum number of samples are misclassified. Considering a set of features \(f_j, j = 1, \ldots, M\), the weak hypothesis depends on only a single feature \(f_j\). On each round \(t\), a weak hypothesis \(h_j\) consists of a feature \(f_j\), a threshold \(\theta_j\) and a parity \(p_j\) (either \(-1\) or \(1\)) indicating the direction of the inequality sign:

\[
h_j(x) = \begin{cases} 
1, & \text{if } p_j f_j(x) < p_j \theta_j, \\
0, & \text{otherwise.}
\end{cases}
\]

(1)

The weak learner algorithm finds the optimal values of \(\theta_j\) and \(p_j\) such that the number of misclassified training samples is minimized as:

\[
(p_j, \theta_j) = \arg\min_{(p, \theta)} \sum_{i=1}^{N} D_t(i) |h_j(x_i) - y_i|.
\]

(2)

At the end of each round \(t\), the distribution \(D_t\) is updated in order to increase the importance of samples which were incorrectly classified by the previous selected weak hypothesis. The final strong classifier is a linear combination of the \(T\) weak hypotheses where the weights are inversely proportional to the training errors. The Viola-Jones’ classifier can be viewed as single node decision trees, where these structures are also called decision stumps in the machine learning literature.

The Viola-Jones AdaBoost uses the Classic AdaBoost’ structure. Therefore, the Viola-Jones AdaBoost does not consider the improvements presented by the Generalized AdaBoost in relation to the Classic AdaBoost. To include the effectiveness of both the Viola-Jones AdaBoost and the Generalized AdaBoost, Mozos [21] presented an AdaBoost called Generalized AdaBoost for the place categorization problem using the weak learning algorithm of the Viola-Jones AdaBoost, as specified in Algorithm 2.

Considering a sequence of input samples \((x_1, y_1), \ldots, (x_N, y_N)\), where \(x_i\) belongs to some domain space \(X\) and \(y_i\) belongs to the label set \(Y = \{-1, +1\}\); and \(y_i = +1\) indicates a positive sample \(x_i\) and \(y_i = -1\) indicates a negative sample \(x_i\). Similarly to the other AdaBoosts, the distribution \(D_t\) indicates the importance (weights) of each sample. On each round \(t\), the distribution is normalized and then a weak classifier is selected according to the distribution \(D_t\).

Similarly to the Viola-Jonas AdaBoost, for each feature \(f_j\), a weak classifier of the following form is selected:

\[
h_j(x) = \begin{cases} 
+1, & \text{if } p_j f_j(x) < p_j \theta_j, \\
-1, & \text{otherwise.}
\end{cases}
\]

(3)

Equation (3) differs from Equation (1) in the output for a negative classification. The weak learner algorithm finds the optimal values of \(\theta_j\) and \(p_j\) so as to maximize \(|r_j|\) (see Algorithm 2). Then the distribution \(D_{t+1}\) is updated and the weight of a weak hypothesis \(\alpha_t\) is calculated. After \(T\) rounds, the final classifier is composed by a weighted combination of the chosen weak hypotheses. In the following experiments, Generalized AdaBoost corresponds to the AdaBoost proposed by Mozos [21].

B. Gentle AdaBoost and Modest AdaBoost

This work introduces the Gentle AdaBoost [16] and the Modest AdaBoost [17] algorithms for place categorization. Results indicate that these classifiers perform better in image classification than other classifiers based on Support Vector Machines and Neural Networks [18].

Gentle AdaBoost optimizes the performance of the final classifier using Newton stepping for minimizing the criterion

\[
E[\exp(-yH(x))],
\]

where \(E\) represents the expectation. The algorithm updates a weak hypothesis using class probabilities as:

\[
h_t(x) = P_{D_t}(y = +1|x) - P_{D_t}(y = -1|x).
\]

(4)

Empirical evidence suggests that Gentle AdaBoost has similar classification performance when compared with other AdaBoosts, such as, Real AdaBoost and LogitBoost, but it often outperforms them in terms of stability [16]. The Gentle AdaBoost is outlined in Algorithm 3, where the weak learner is
Algorithm 3 Gentle AdaBoost.
1. Given samples \((x_i, y_i), i = 1, \ldots, N\), where \(y_i = -1\) for negative samples and \(y_i = +1\) for positive samples;
2. Initialize a distribution as \(D_1(i) = \frac{1}{N}, i = 1, \ldots, N\);
3. Set \(t \leftarrow 1\) and \(\mathcal{H}(x) \leftarrow 0\);
4. Repeat
   (a) Train a weak hypothesis \(h_t(x)\) using distribution \(D_t(i)\) by weighted least squares;
   (b) Update \(\mathcal{H}(x) \leftarrow \mathcal{H}(x) + h_t(x)\);
   (c) Update the distribution as:
       \[ D_{t+1}(i) = D_t(i) \exp(-y_i h_t(x_i)) \]
   (d) Normalize the distribution so that:
       \[ \sum_{i=1}^{N} D_{t+1}(i) = 1 \]
   (e) Set \(t \leftarrow t + 1\); until \(t = T\);
5. Construct the final classifier as:
   \[ \mathcal{H}(x) = \text{sign}(F(x)), \text{where } F(x) = \sum_{t=1}^{T} h_t(x). \]

Fitted by weighted least squares as:
\[ h_t(x) = \arg\min_{h} \left( \sum_{i=1}^{N} D_t(i) \cdot (y_i - h(x_i))^2 \right). \quad (5) \]

Modest AdaBoost has less generalization error and higher training errors when compared to the other AdaBoosts. Experiments show that Modest AdaBoost outperforms Gentle AdaBoost in terms of generalization error, but reduces training error much slower, sometimes not reaching zero [17]. A description of Modest AdaBoost is shown in Algorithm 4. The only requirement for a weak hypothesis \(h_t\) is that for any of its output values, the algorithm should be able to estimate probability \(P_{D_t}(y = +1 | h_t(x))\).

At each round \(t\), distribution \(D_t\) assigns high weights to training samples misclassified in earlier stages. On the contrary, the inverse distribution \(\bar{D}_t\) gives higher weights to samples correctly classified in earlier rounds. The initial weights are set as \(D_1(i) = \frac{1}{N}\). The new distribution \(\bar{D}_{t+1}\) is calculated as:
\[ \bar{D}_{t+1}(i) = \frac{D_t(i) \exp(-y_i g_t(x_i))}{Z_t}, \quad (6) \]

where \(Z_t\) is a normalization coefficient. \(Z_t\) and \(\bar{Z}_t\) are chosen so that:
\[ \sum_{i=1}^{N} \bar{D}_t(i) = \sum_{i=1}^{N} D_t(i) = 1. \quad (7) \]

A weaker learner is fitted in Step 4a by weighted least squares according to the Equation 5.

Expressions \(P_t^{+1}(x) = P_{D_t}(y = +1 | h_t(x))\) and \(P_t^{-1}(x) = P_{D_t}(y = -1 | h_t(x))\) compute how good the current weak hypothesis is predicting the class labels. Expressions \(\bar{P}_t^{+1}(x) = P_{\bar{D}_t}(y = +1 | h_t(x))\) and \(\bar{P}_t^{-1}(x) = P_{\bar{D}_t}(y = -1 | h_t(x))\) estimate how good the current weak hypothesis is working on the data which has been correctly classified by previous steps. So when Algorithm 4 uses \(g_t(x) = (P_t^{+1}(1 - \bar{P}_t^{+1}) - P_t^{-1}(1 - \bar{P}_t^{-1}))(x)\) as an update for current step, it decreases the weak hypothesis contribution, if it works “too good” on data that has already been correctly classified with high margin. A feature of the Modest AdaBoost is that \(1 - \bar{P}_t^y, y \in \{-1, +1\}\) can actually become zero, so then update will not occur. This provides a natural stopping criterion.

IV. Multi-class Classification Using a Probabilistic Decision List of AdaBoost Classifiers

The AdaBoost algorithms presented in Section III have been designed for binary classification problems. However, in this work, it is necessary to handle multiple classes for place categorization. To cover this issue, Mozos et al. [15] shows that it is possible to design an AdaBoost arranged into a Decision List (DL) for multi-class problems, where each AdaBoost of the DL determines if a sample belongs to one specific class. The aim is to train each AdaBoost using samples of its class as positives and the other samples as negatives. Consider a DL that is defined by a sequence of \(K\) classes \(k = 1, \ldots, K\),
Figure 1: Probabilistic decision list using binary AdaBoosts.

and by $K-1$ AdaBoosts. An AdaBoost uses a binary hypothesis $h_k : X \to Y = \{0, 1\}$ to classify a sample $x$. Each hypothesis $h_k$ determines if a sample $x$ belongs to the corresponding class $k$. If an AdaBoost gives output 1, then the sample is assigned to class $k$. On the contrary, the sample $x$ is passed to the next AdaBoost into the list until it is attributed to a class.

This idea can be extended to a PDL, where each AdaBoost gives a confidence value $C_j^k$ for its class $[21]$, as shown in Figure 1. The sample is passed to the next AdaBoost, but with a negative classification confidence value of $1 - C^+$, where:

$$ C^+ = P(y = +1|x) = \frac{e^{F(x)}}{e^{-F(x)} + e^{F(x)}}, \quad (8) $$
$$ C^- = P(y = -1|x) = \frac{e^{-F(x)}}{e^{-F(x)} + e^{F(x)}}, \quad (9) $$
$$ C^+ = P(y = +1|x) = 1 - C^- \quad (10) $$

The complete output of a sample $i$ can be defined as a histogram $P_i$, where each bin $p_i(k)$ of $P_i$ stores the probability of a class $k$:

$$ p_i(k) = C_k^+ \prod_{j=1}^{K-1} (1 - C_j^+), \quad (11) $$

where $\sum_{k=1}^{K} p_i(k) = 1$ and the confidence value $C_K^+$ of the last bin $p_i(K)$ is equal to $C_K = 1$ according to the PDL structure.

Another important issue of the DL is the order in which the AdaBoosts are arranged, because the individual classifiers may have different performances. The experiments aim to find the optimal order of the AdaBoosts into the PDL.

V. Probabilistic Relaxation Labeling

Relaxation labeling was first proposed by Rosenfeld et al. [22]. The method employs contextual information to classify a set of interdependent objects by allowing interactions among the possible classifications of the related objects. Relaxation labeling has been applied to many areas of the computer vision [23] and for smoothing the final classification of the semantic maps [15]. In this last case, the label that is attributed to each cell of the map depends on the labels in its neighborhood.

This work applies Probabilistic Relaxation Labeling (PRL) for smoothing the classification of the PDL as suggested by Mozos et al. [15]. The aim is to classify the labels of the unoccupied cells of the maps. Let us consider a labeling problem with a set of $B$ cells $C = \{c_1, \ldots, c_B\}$, and $K$ labels $Y = \{y_1, \ldots, y_K\}$. The labeling is defined by a function that maps each element of $C$ into an element of $Y$. Each cell $c_i$ is related to a probabilistic distribution given by the histogram $P_i$ (as shown in Section IV), where each bin $p_i(k)$ of the histogram stores the probability that the cell $c_i$ has the label $k$.

Let $Ne(c_i)$ be the set of cells that influence the labeling process of cell $c_i$ by PRL. Here $Ne(c_i)$ is defined to be the neighborhood of $c_i$, which consists of the cells $c_j \neq c_i$ that are connected to $c_i$. Here the type of connectivity between a cell $c_i$ with coordinates $(x, y)$ and its neighbors is 8-connectivity. Thus, each cell in the interior of the state-space has 8 neighboring cells.

For each pair of cells $(c_i, c_j)$, $d_{ij}$ denotes the influence on $c_i$ from $c_j$, $D = \{d_{ij} | c_j \in Ne(c_i)\}$, $\sum_{j=1}^{Q} d_{ij} = 1$, and $Q_i$ is the number of neighbors of $c_i$. The compatibility between the label $k$ of cell $c_i$ and the label $k'$ of cell $c_j$ is given by $r_{ij}(k,k') \in [-1, 1]$, satisfying the condition:

$$ r_{ij}(k,k') = \frac{1 - p_i(k)}{p_i(k)}, \quad (12) $$

and $R = \{r_{ij}(k,k')|c_j \in Ne(c_i)\}$. High values of $r_{ij}(k,k')$ correspond to compatibility and low values to incompatibility. The term $p_{ij}(k|k')$ is the conditional probability that cell $c_i$ has label $k$ given that cell $c_j \in Ne(c_i)$ has label $k'$. This paper calculates $p_{ij}(k|k')$ by:

$$ p_{ij}(k|k') = \frac{p_{ij}(k \cap k')}{{p_i(k')}, \quad (13) $$

where $p_{ij}(k \cap k')$ is the joint probability, i.e. the probability of labels $k$ and $k'$ occurring simultaneously; and $p_i(k')$ is the probability that cell $c_j$ has label $k'$. The compatibility coefficients $r_{ij}(k,k')$, $p_i(k)$ and $p_{ij}(k|k')$ maintain the same values during the PRL process. They are calculated using the training set. For each cell $c_i$, there is a set of initial probabilities given by $P_i^{(0)} = \{P_i^{(0)}(k)|k = 1, \ldots, K\}$, satisfying $\sum_{k=1}^{K} P_i^{(0)}(k) = 1$. These probabilities are given by the PDL of AdaBoosts.

PRL computes the histogram $P_i$ assigned to each cell $c_i$ iteratively until the labeling method converges or stabilizes. The updated probability of having label $k$ on $c_i$ at the $(t+1)$-th iteration is:

$$ p_i^{(t+1)}(k) = \frac{p_i^{(t)}(k) \left[ 1 + q_i^{(t)}(k) \right]}{\sum_{k'=1}^{K} p_i^{(t)}(k') \left[ 1 + q_i^{(t)}(k') \right]}, \quad (14) $$

where $q_i^{(t)}(k) = \sum_{j=1}^{N} d_{ij} \left( \sum_{k'=1}^{K} r_{ij}(k,k') p_j^{(t)}(k') \right). \quad (15)$

The key issue of the PRL process is to define how many iterations are needed. In this work, PRL is performed $T$ times and the iteration $t$ with the lowest classification error is taken as the solution for the PRL process. The weights $d_{ij}$ are set with value $\frac{1}{Q}$, i.e. all the cells in the neighborhood of $c_i$ have same importance.

In the occupancy grid map process, we may be dealing with occupied cells (i.e. cells defined as wall) in the neighborhood of a free cell $c_i$. In this case, in the PRL process wall
must be included in the set of labels and the wall probabilities must be defined. The initial probability $p_i^{(0)}(\text{wall})$ of a cell $c_i$ with label wall is set as 1; and other initial probabilities i.e. $p_i^{(0)}(\text{room})$, $p_i^{(0)}(\text{corridor})$ and $p_i^{(0)}(\text{doorway})$ are set as 0. For a free cell $c_i$ (i.e. room, corridor or doorway), the initial probabilities $p_i^{(0)}(\text{room})$, $p_i^{(0)}(\text{corridor})$ and $p_i^{(0)}(\text{doorway})$ correspond to those given by the PDL, and $p_i^{(0)}(\text{wall})$ is set as 0.

VI. Heuristics for Removing Incorrect Regions

Some heuristics are necessary for removing incorrect regions after the relaxation labeling process. The aim is to increase the classification rate of the maps [21].

Connected component (or region) labeling references to the task of grouping the connected pixels with the same label in an image, where each region is labeled based on a given method. Connected components are extracted using 8-connected cells with same class $y \in \{\text{room, corridor, doorway}\}$ [24]. The label of a region is the same as the label of its pixels. Moreover, for each region, its connection (i.e. border), area and number of cells are extracted. The heuristics proposed for removing incorrect! regions are defined as follows:

1. Remove regions of type room or corridor whose number of cells is lower than a given threshold which is set to 400 in this paper. For each removed region, its cells are merged into a new cumulative region which also includes all neighboring regions of type $w$, where $w$ is the most frequent type of pixel in the border of the removed region;

2. Remove regions of type doorway that are connected to only one label. For each removed region, its cells are merged into a new cumulative region which also includes all neighboring regions. (Note: each region of type doorway connected to both room and corridor is kept even if its number of cells is less than a threshold).

VII. Experiment Results

Below the experiments use the data sets available on Óscar Martínez Mozos’s website\(^1\). They correspond to the Building 52 and Building 79 at the University of Freiburg. Both data sets have 150 features, as described in Table 3, and three classes that represent the places of the Buildings (room, corridor and doorway). The data sets are divided into training data set and testing data set, as shown in Figures 2 and 3.

The aim of the experiments is to demonstrate the improvements obtained by the PDL of AdaBoosts when the described PRL process and the proposed heuristics are performed. The tested AdaBoosts include the Viola-Jones AdaBoost, the Generalized AdaBoost, the Gentle AdaBoost and the Modest AdaBoost.

The Viola-Jones AdaBoost and the Generalized AdaBoost were implemented according to Algorithms 1 and 2, respectively. In this case, for each weak learner $h_j$ based on a single feature $f_j$, a set of thresholds $b_{ij}$ is generated in a linearly

---

\(^1\)http://www.informatik.uni-freiburg.de/~omartine/

\(^2\)http://graphics.cs.msu.ru/ru/science/research/machinelearning/adaboosttoolbox

Figure 2: Map of the building 52 - (a) training data set; (b) testing data set; room, corridor, doorway.

Figure 3: Map of the building 79 - (a) training data set; (b) testing data set; room, corridor, doorway.
and the Modest AdaBoost) has the best order.

B. Experiments Using the Building 52 and the Building 79 Data Sets

The methodology applied in the experiments can be divided into three steps. In the first step, a PDL of AdaBoosts with the best order is built, where the AdaBoosts are trained using the training data set. Then, the PDL is used to classify the cells of the testing map; and it outputs the histogram $P_i$ of each cell $c_i$. In the second step, the classifications are smoothed using the PRL process presented in Section V. This method receives as inputs the histograms $P_i$, the conditional probabilities $p_{ij}(k|k')$, the initial probabilities $p_i^{(0)}(k)$ and the maximum number of iterations $T$ (set as 50); and it outputs a testing map with a smoothed distribution of classification. In the third step, some incorrectly classified regions are re-classified using the heuristics described in Section VI. Table 5 shows the experimental results obtained on Building 52 and Building 79 using the PDL of AdaBoosts. The column $Ord.$ specifies the best order of the AdaBoosts according to Table 4. The classification rates measure the percentage of samples correctly classified, where \( train_{52} \) is the rate on the training data set and \( test_{52} \) is the rate on the testing data set. \( AdaBoost_1 \), and \( AdaBoost_2 \) are the first and second AdaBoosts of the PDL, respectively; PDL shows the classification rates using a PDL with the \( AdaBoost_1 \) and \( AdaBoost_2 \); PRL reports the classification rate after having applied probabilistic relaxation labeling using the best number of iterations $t$; Heur. shows the classification rate after having performed the proposed heuristics on the test map. The classification rates indicate that the Gentle AdaBoost has the best performance for the \( AdaBoost_1 \) and the \( AdaBoost_2 \) in the training data set. In the testing data set for the \( AdaBoost_1 \) and the \( AdaBoost_2 \), the Generalized AdaBoost has worse classification rates when compared to the other AdaBoosts. The best classification rate with a PDL in the testing data is obtained to the Gentle AdaBoost in the case of Building 52 and by Modest AdaBoost in the case of Building 79. When the relaxation labeling and the heuristics are applied, the Generalized AdaBoost significantly improves its performance.

C. Maps of the Building 52 and the Building 79

This Subsection presents the final maps using the classification outputs for the methodologies described above. Figures 4 and 5 show the maps obtained for Building 52 and Building 79 at the University of Freiburg, respectively. As displayed, the environments are divided as rooms, corridors and doorways. After applying the PDL of AdaBoosts to classify the places, some noises can be seen in the Figures. These noises are mainly due to cells misclassified as doorways, for example, as shown in the Figure 4d. When the relaxation labeling is applied, a reduction of noise can be easily observed. In the other cases, for example in Figure 4k, it is noticed the plausible arising of cells with the doorway label. However, even applying the relaxation labeling, some incorrect regions are presented by the maps, as can be seen clearly in Figure 4e. Later these regions are eliminated using the heuristics, as shown in Figure 4f.

In general, the major problem is to classify correctly the doorways. The Generalized AdaBoost has the best performance on the doorways when compared to the other AdaBoosts. For Building 79, the Viola-Jones AdaBoost has a performance similar to the Generalized AdaBoost. The Modest AdaBoost fails in the map of Building 79. However, for the map of Building 52, the relaxation labeling strongly improves the classification rate of the doorways. This idea can be also extended to other learning algorithms.

D. Comparison with the Literature Results

Different AdaBoosts integrated with PDL for semantic place labeling in indoor environments are proposed. The tests indicate that the best AdaBoosts for place categorization are the Generalized AdaBoost and the Viola-Jones AdaBoost, where the final classification rate is 98.90% for Building 52 and 99.16% for Building 79. Mozos [21] also presents the PDL of Generalized AdaBoost, relaxation labeling, and heuristics for removing incorrect regions. Their final results are slightly inferior when compared to the results presented here, where their classification rate are 98.66% for Building 52 and 98.95% for Building 79. Here, the better results are attributed to the choice of the iteration $t$ with lowest classification error on the iterations $T$ as solution of the PRL process.

VIII. Conclusions

This paper compares binary AdaBoosts integrated with PDL for solving multi-class problems. The aim is to classify places in indoor environments using a mobile robot equipped with a laser range scanner covering 360° field of view. To increase the classification rate, probabilistic relaxation labeling and some heuristics are applied. The experiments are based on the data sets acquired from both the Building 52 and the Building 79 at the University of Freiburg. In these buildings, the environments are divided into places of three types: room, door and corridor.

The investigated AdaBoosts are the Viola-Jones AdaBoost, the Generalized AdaBoost, the Gentle AdaBoost and the Modest AdaBoost. The experimental results have shown that PDL of Generalized AdaBoost has inferior classification rate when compared to PDLs of other AdaBoosts. However, in Building 52, the Generalized AdaBoost outperforms the other AdaBoosts when relaxation labeling and heuristics are performed; And, in Building 79, the Viola-Jones AdaBoost has the best performance.

In Building 52, the tests have shown that the optimal PDL is the one composed of Generalized AdaBoosts, where the final classification rate in the test map is 98.90%; And the experimental results on Building 79 have shown that the best PDL...
Figure 4: Test map of the building 52 using PDL of: (a)-(c) Viola-Jones AdaBoost; (d)-(f) Generalized AdaBoost; (g)-(i) Gentle AdaBoost; (j)-(l) Modest AdaBoost; room, corridor, doorway.

Table 5: Experimental results using the AdaBoosts into the probabilistic decision list.

<table>
<thead>
<tr>
<th>AdaBoost Algorithm</th>
<th>Order</th>
<th>AdaBoost$_1$</th>
<th>AdaBoost$_2$</th>
<th>PDL</th>
<th>PRL</th>
<th>Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>train$_%$</td>
<td>test$_%$</td>
<td>train$_%$</td>
<td>test$_%$</td>
<td>train$_%$</td>
</tr>
<tr>
<td>Viola-Jones</td>
<td>3</td>
<td>98.42</td>
<td>98.16</td>
<td>99.94</td>
<td>99.66</td>
<td>98.36</td>
</tr>
<tr>
<td>Generalized</td>
<td>2</td>
<td>99.61</td>
<td>99.35</td>
<td>95.60</td>
<td>95.67</td>
<td>95.88</td>
</tr>
<tr>
<td>Gentle</td>
<td>3</td>
<td>99.79</td>
<td>98.08</td>
<td>99.81</td>
<td>99.88</td>
<td>99.79</td>
</tr>
<tr>
<td>Modest</td>
<td>2</td>
<td>99.70</td>
<td>99.58</td>
<td>97.73</td>
<td>97.72</td>
<td>97.79</td>
</tr>
</tbody>
</table>

The Building 79 at the University of Freiburg

<table>
<thead>
<tr>
<th>AdaBoost Algorithm</th>
<th>Order</th>
<th>AdaBoost$_1$</th>
<th>AdaBoost$_2$</th>
<th>PDL</th>
<th>PRL</th>
<th>Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>train$_%$</td>
<td>test$_%$</td>
<td>train$_%$</td>
<td>test$_%$</td>
<td>train$_%$</td>
</tr>
<tr>
<td>Viola-Jones</td>
<td>1</td>
<td>98.59</td>
<td>99.13</td>
<td>97.98</td>
<td>98.10</td>
<td>98.10</td>
</tr>
<tr>
<td>Generalized</td>
<td>1</td>
<td>98.56</td>
<td>98.59</td>
<td>96.53</td>
<td>97.63</td>
<td>97.63</td>
</tr>
<tr>
<td>Gentle</td>
<td>2</td>
<td>99.99</td>
<td>99.45</td>
<td>99.53</td>
<td>98.12</td>
<td>99.57</td>
</tr>
<tr>
<td>Modest</td>
<td>2</td>
<td>99.13</td>
<td>99.39</td>
<td>98.18</td>
<td>99.32</td>
<td>97.80</td>
</tr>
</tbody>
</table>
is also the one composed by Viola-Jones AdaBoosts, where the classification rate of the test map was 99.16%. The results suggested that PDL can be extended to several binary classifiers and show that PRL improves significantly the classification rates of the classifiers.

Acknowledgments

Symone Soares is supported by the Fundação para a Ciência e a Tecnologia (FCT) under the Grant SFRH/BD/68515/2010. The authors acknowledge the support of FCT project PEst-C/EEI/UI0048/2013.

References


Author Biographies

Symone G. Soares received her graduation degree in Computer Engineering from the Pontifical Catholic University of Goiás (PUC-GO), Brazil, in 2009. Actually she is pursuing her Ph.D degree in Electrical and Computer Engineering at the University of Coimbra, Portugal. Since 2010, she is a researcher at the Portuguese Institute for Systems and Robotics (ISR-Coimbra). Her research interests include machine learning, optimization techniques, computational intelligence modeling, and Soft Sensors for industrial applications.
**Rui Araújo** received the B.Sc. degree ("Licenciatura") in Electrical Engineering, the M.Sc. degree in Systems and Automation, and the Ph.D. degree in Electrical Engineering from the University of Coimbra, Portugal, in 1991, 1994, and 2000 respectively. He joined the Department of Electrical and Computer Engineering of the University of Coimbra where he is currently an Assistant Professor. He is a founding member of the Portuguese Institute for Systems and Robotics (ISR-Coimbra), where he is now a researcher. His research interests include computational intelligence, intelligent control, learning systems, fuzzy systems, neural networks, control, embedded systems, real-time systems, soft sensors, industrial systems, sensor-based mobile robot navigation, and in general architectures and systems for controlling robot manipulators, and for controlling mobile robots.