Tackling Fingerprinting Indoor Localization Using the LASSO and the Conjugate Gradient Algorithms

Matheus A. Marins, Rafael S. Chaves, Vinicius M. de Pinho, Rebeca A. F. Cunha, and Marcello L. R. de Campos

***Abstract*— This paper presents the application and the comparison of the least absolute shrinkage and selection operator (LASSO) and the Conjugate Gradient (CG) algorithm for solving the fingerprinting indoor localization problem. LASSO's ability to generate sparsity via selection of variables results in a judicious and automatic removal of spurious measurements that often corrupt large fingerprint data sets. These spurious measurements usually have to be individually discarded before the CG algorithm, or other solver for the normal equation, is used. The paper also compares LASSO with a sparse version of the ordinary least squares solution obtained by simply discarding the variables with the smallest absolute value. The results are presented for two data sets recorded independently at the State University of Rio de Janeiro, Brazil, and at Universitat Jaume I, located between the cities of Valencia and Barcelona, Spain.**

***Keywords*— WLAN fingerprinting, indoor localization, Conjugate Gradient, LASSO**

I. Introduction

Ordering a cab, interacting in a social network, and even participating in online multiplayer games which create different and realistic scenarios are examples of applications benefited from location information. Everyday more location-based services (LBS) are developed increasing the demand for better solutions of the indoor localization problem, where the global positioning system (GPS) fails to provide service. Modern devices are expected to obtain, utilize, and sometimes share their positions, but technologies based on Bluetooth [1], radio frequency identification (RFID) [2], [3], and specially designed radio-frequency setups [4] are usually expensive, or computationally demanding. As better usually means cheaper and ubiquitous, successful approaches must be accurate and at the same time shall require minimum additional deployment and minimum energy expenditure.

The technique described herein uses the already existent wireless local area network (WLAN) infrastructure to provide an inexpensive alternative to indoor localization. Position is determined as a linear combination of the different received signal strength indicators (RSSI) available from WLAN access points at a particular location. The coefficients are estimated beforehand solving a least squares (LS) problem using measurements collected in the area. In a previous work [5], we showed the application of the conjugate gradient (CG) algorithm to obtain the LS solution, and ascertained the necessity of careful conditioning the data in order to remove spurious measurements. In this work, we maintain that a sparsity promoting strategy for solving the optimization cost function, such as the least absolute shrinkage and selection operator (LASSO), yields very good results and is able to discard otherwise harmful measurements automatically. Furthermore, this fingerprinting approach has shown to provide better accuracy than range-based techniques [6].

The paper is organized as follows. In Section II, we introduce the indoor localization problem and how it is parameterized. Section III presents the CG algorithm and explains how the localization problem is solved. Section IV presents the LASSO algorithm. Section V presents the results of experiments using real data collected in two different locations in order to analyze and compare the performance of the CG and LASSO algorithms. Finally, some conclusions are drawn in Section VI.

Notation: Vectors and matrices are represented in bold face with lowercase and uppercase letters, respectively. The notations $[\cdot]^T$, $[\cdot]^{-1}$ stand for transpose, and inverse operations on $[\cdot]$, respectively. The element in the i-th row and j-th column of matrix $\mathbf{A}$ is given by a_{ij}. The symbol $\mathbb{R}$ denotes the set of real numbers. The set of $\mathbb{R}^{M\times N}$ denotes all $M\times N$ matrices comprised of real-valued entries. The symbols $\mathbf{1}_{M\times N}$ and $\mathbf{I}_M$ denote an $M\times N$ matrix with ones and the $M\times M$ identity matrix, respectively.

II. Indoor Localization Using Fingerprinting

Fingerprinting localization consists of two phases: offline fingerprinting calibration and online location estimation. In the offline phase, a set of RSSI values is collected from various access points to form one of the fingerprints stored in the database. The procedure is repeated for several different locations forming a dense grid of reference points, each one associated with a fingerprint. The challenge here is to construct an up-to-date RSSI map with reasonable labor cost in this phase. During the online phase, the location of the terminal is obtained from comparing the fingerprint collected in real-time with those in the database.

While many solutions use the data in matching algorithms, like k-nearest neighbor [6] and artificial neural networks [7], in this paper the database is used to build a system of M affine equations with N unknowns. The i-th equation is given by

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{iN}x_N = b_i, \tag{1}$$

where $a_{ij}\in\mathbb{R}$ stands for the received RSSI of the j-th access point (AP) at position b_i. These set of M equations can be written in a matrix form as

$$\mathbf{Ax} = \mathbf{b}, \tag{2}$$

where $\mathbf{A}\in\mathbb{R}^{M\times N}$ is expected to be a full column rank matrix, $\mathbf{x}\in\mathbb{R}^{N\times 1}$ and, $\mathbf{b}\in\mathbb{R}^{M\times 1}$. Even when the system does not

Matheus A. Marins, Rafael S. Chaves, Vinicius M. de Pinho, Rebeca A. F. Cunha, and Marcello L. R. de Campos, are with the Department of Electronics and Computer Engineering, Poli/Federal University of Rio de Janeiro, and the Electrical Engineering Program, COPPE/Federal University of Rio de Janeiro, Rio de Janeiro, RJ 21941-972, Brazil (e-mails: {matheus.marins, rafael.chaves, vinicius.pinho, rebeca.araripe, campos}@smt.ufrj.br).

have a unique solution, one may be interested in finding $\hat{\mathbf{x}}$ that minimizes the optimization problem given by

$$\min_{\mathbf{x}} \; f(\mathbf{x}) = \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2. \tag{3}$$

The equation (3) represents the OLS (Ordinary Least Square), and its solution can be obtained through the normal equation,

$$\mathbf{A}^T\mathbf{A}\hat{\mathbf{x}} = \mathbf{A}^T\mathbf{b}. \tag{4}$$

In practice, $\mathbf{A}$ may not have full column rank and the solution to equation (3) may require careful, and sometimes tedious, manipulation of data before equation (4) is solved for $\hat{\mathbf{x}}$. Once the vector of weights $\hat{\mathbf{x}}$ is computed, one can obtain the coordinates of new locations by applying it to a new matrix of RSSI collected at those locations.

III. Conjugate Gradient Method

The Conjugate Gradient is an iterative method that can be used to solve quadratic optimization problems, such as the problem in presented in equation (3):

$$\begin{aligned}\min_{\mathbf{x}} \; f(\mathbf{x}) &= \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \\ &= \frac{1}{2}\mathbf{x}^T\mathbf{F}\mathbf{x} - \mathbf{q}^T\mathbf{x} + \|\mathbf{b}\|_2^2,\end{aligned} \tag{5}$$

where $\mathbf{F} = 2\mathbf{A}^T\mathbf{A}$, $\mathbf{F} \in \mathbb{R}^{N\times N}$ e $\mathbf{q} = 2\mathbf{A}^T\mathbf{b}$, $\mathbf{q} \in \mathbb{R}^{N\times 1}$. The CG algorithm produces the optimal solution after N iterations in the absence of round-off errors [8]. If $\mathbf{F}$ is sparse, solving problems like equation (5) with the CG algorithm is faster than solving equation (4) using direct methods.

The CG algorithm uses the following recursive equation

$$\mathbf{x}_{(k+1)} = \mathbf{x}_{(k)} + \alpha_{(k)}\mathbf{d}_{(k)}, \tag{6}$$

where $\alpha_{(k)} \in \mathbb{R}$ and $\mathbf{d}_{(k)} \in \mathbb{R}^{N\times 1}$ are the step size and the direction in iteration k, respectively. The core difference between CG and other gradient methods, such as the steepest decent (SD) [9], is the choice of step size and direction. At each iteration, the CG method takes a step in the direction $\mathbf{d}_{(k)}$, $\mathbf{F}$-*conjugate* (or $\mathbf{F}$-*orthogonal*) to all the previous choices. The step size, which is controlled by parameter $\alpha_{(k)}$, must be chosen so that it minimizes $f(\mathbf{x}_{(k)})$ along $\mathbf{d}_{(k)}$, hence $\alpha_{(k)}$ is given by [10]

$$\alpha_{(k)} = -\frac{\nabla_{\mathbf{x}} f(\mathbf{x}_{(k)})^T\mathbf{d}_{(k)}}{\mathbf{d}_{(k)}^T\mathbf{F}\mathbf{d}_{(k)}} \tag{7}$$

and

$$\mathbf{d}_{(k+1)} = -\nabla_{\mathbf{x}} f(\mathbf{x}_{(k+1)}) + \beta_{(k)}\mathbf{d}_{(k)}, \tag{8}$$

where

$$\beta_{(k)} = -\frac{\nabla_{\mathbf{x}} f(\mathbf{x}_{(k+1)})^T\mathbf{F}\mathbf{d}_{(k)}}{\mathbf{d}_{(k)}^T\mathbf{F}\mathbf{d}_{(k)}}. \tag{9}$$

The residue, $\mathbf{r}_{(k)} = \mathbf{q} - \mathbf{F}\mathbf{x}_{(k)}$, and the gradient of the function have opposite directions, thus we can write $\mathbf{r}_{(k+1)}^T\mathbf{d}_{(k)} = 0$, which is equivalent to $\mathbf{d}_{(k)}^T\mathbf{F}\mathbf{e}_{(k+1)} = 0$, where $\mathbf{e}_{(k)}$ is the error from each iteration, $\mathbf{e}_{(k)} = \hat{\mathbf{x}} - \mathbf{x}_{(k)}$. The CG method eliminates the error component in each direction $\mathbf{d}_{(k)}$, revealing an advantage when compared to the SD method, which eventually repeats the optimization along directions already visited, slowing down convergence. This advantage can be seen in Fig. 1.

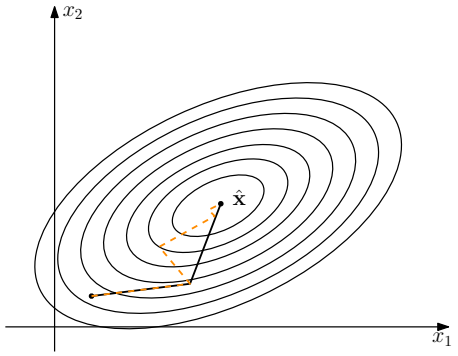

Fig. 1. Search paths of the steepest decent and conjugate gradient methods.

The set of $\mathbf{F}$-*conjugate* search directions is built from the residues with the Gram-Schmidt conjugation process, which eliminates components that are not $\mathbf{F}$-*conjugate* to all the past search directions. Consequently, the subspace generated is the same as the one composed of the residues, i.e.,

$$\mathscr{R}_k = \text{span}\left\{\mathbf{r}_{(0)}, \mathbf{r}_{(1)}, \cdots, \mathbf{r}_{(k-1)}\right\} \tag{10}$$

is equivalent to

$$\mathscr{D}_k = \text{span}\left\{\mathbf{d}_{(0)}, \mathbf{d}_{(1)}, \cdots, \mathbf{d}_{(k-1)}\right\}. \tag{11}$$

Since $\mathbf{r}_{(k+1)} = \mathbf{F}\mathbf{e}_{(k+1)} = \mathbf{r}_{(k)} - \alpha_{(k)}\mathbf{F}\mathbf{d}_{(k)}$, it is clear that the residue $\mathbf{r}_{(k+1)}$ is a linear combination of the residue $\mathbf{r}_{(k)}$ with the previous search directions transformed by $\mathbf{F}$, denoted here as $\mathbf{F}\mathscr{D}_k$. Therefore, the subspace $\mathscr{D}_{k+1}$ is a union of $\mathscr{D}_k$ with $\mathbf{F}\mathscr{D}_k$, and the Krylov subspace is given by

$$\mathscr{D}_{k+1} = \text{span}\left\{\mathbf{d}_{(0)}, \mathbf{F}\mathbf{d}_{(0)}, \mathbf{F}^2\mathbf{d}_{(0)}, ..., \mathbf{F}^k\mathbf{d}_{(0)}\right\}. \tag{12}$$

The fact that $\mathbf{r}_{(k+1)}$ is always orthogonal to all previous residues implies that it is also orthogonal to the subspace constructed with the previous directions. Due to the nature of formation of Krylov subspaces, $\mathbf{r}_{(k+1)}$ is $\mathbf{F}$-*conjugate* to all the past search directions, except the one immediately before the current direction. Therefore, there is no need to store all previous search directions for calculating the Gram-Schmidt conjugation, which is an advantage of the Conjugate Gradient when compared with other methods.

IV. The LASSO Algorithm

When $\mathbf{A}$ is not a full column rank matrix, removing the linearly dependent columns to transform the measurement matrix $\mathbf{A}$ in a full rank matrix is one way to approach the problem. This procedure may require intervention of the designer, and the quality of the outcome will depend on his or her experience and skills. Another approach is to impose constraints to the cost function. This is usually done by limiting some norm of the solution, such as the l_2-norm of $\mathbf{x}$, which yields the ridge regression (or Tikhonov regularization) [11].

The ridge regression is mathematically described by

$$\min_{\mathbf{x}} \; \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_2^2, \;\; \lambda \geq 0, \tag{13}$$

which leads to

$$\hat{\mathbf{x}} = \left(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{A}^T\mathbf{b}. \tag{14}$$

A suitable choice of λ makes $(\mathbf{A}^T\mathbf{A}+\lambda\mathbf{I}_N) \succ 0$, i.e., positive definite. As a consequence, the solution of equation (13) is unique and is a shrunk version of the solution of equation (3).

Another way to deal with the ill-conditioned equation (3) is imposing an l_1-norm restriction on $\mathbf{x}$, known as LASSO [12]. The LASSO algorithm seeks to find the optimal solution of equation (3), subject to a maximum value of its l_1-norm. The optimization problem can be written as

$$\min_{\mathbf{x}} \ \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \quad \text{s.t. } \|\mathbf{x}\|_1 \leq t. \tag{15}$$

The constraint imposed to the l_1-norm is responsible for the shrinkage of the vector and controls the sparsity of the solution. If $t \geq \sum|\hat{x}_i|$, LASSO will find the ordinary least square (OLS) solution; decreasing the value of t, a more sparse solution arises.

Fig. 2 illustrates a 2-D example of how LASSO works. The center of the ellipse, $\hat{\mathbf{x}}$, is the OLS solution and the elliptical contours are equal-error regions. The hatched area in Fig. 2 is the set of points that satisfy the l_1-norm condition. Decreasing the value of t in equation (15) will promote sparsity in the solution as soon as the elliptical contour touches the l_1-norm region along the axis, as in Fig. 2.

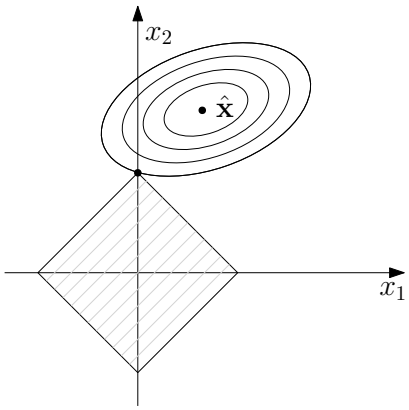

Fig. 2. 2-D representation of LASSO's problem.

The optimization problem of the LASSO algorithm can be written on its dual form as [9]

$$\min_{\mathbf{x}} \ \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2+\lambda\|\mathbf{x}\|_1, \ \lambda \geq 0. \tag{16}$$

The results presented in Section V were generated by an algorithm that solves the dual form of LASSO. Parameter λ is a variable of the dual function. It is a regularization parameter that will vary according to how much the norm l_1 is capped by t. The relation between t and λ depends on the data: if λ increases, more vector elements are shrunk towards zero, whereas when λ is zero, the solution is equivalent to that of the OLS problem.

V. Simulations and Results

A. Databases

The first database used in this paper, called UJIndoorLoc, is presented and fully explained in [13], but a short explanation of some of its parameters is necessary for understanding the results obtained in this paper. The data set is formed by a matrix whose 520 columns have the RSSI information from the 520 different wireless access points (WAP). This matrix has 21048 rows; 19937 of them were used here for the training phase (or offline phase), and 1111 were used for the validation phase (or online phase). Each row represents a measurement at one location, where the latitude and longitude coordinates were also recorded. The place in which the data were collected has a total area of 108703 m^2, which includes 3 buildings with 4 or 5 floors. When no signal from a WAP was received at a measurement point, its RSSI was recorded as 100. The RSSI range is $[-100, 0]$ dBm, in which -100 represents the minimum signal strength possible and 0 the maximum. In our implementation, all entries equal to 100 were replaced by -200.

The second database used was collected at the Reitor João Lyra Filho Pavilion at the Rio de Janeiro State University (UERJ) [14]. The measurements were carried out in 924 different points distributed among the 13 floors of the building. At each point, the RSSIs (ranging from -120 to -30 dBm) received from the 136 existent WAPs were collected, along with their coordinates. In this database, when there was no signal received from a WAP, a $-\infty$ was recorded. Once again, we replaced all these entries for -200. A total of 130702 measurements of this data set were used here, divided in training phase (70%) and validation phase (30%).

B. Conjugate Gradient Method

The algorithm implementation of the CG method used in this paper was based on the pseudocode presented in [10]. For the first data set, equation (5) led to a matrix $\mathbf{F}_1 \in \mathbb{R}^{520\times520}$, whose condition number was 4.1735×10^{38}. In order to eliminate data that hardly provided information, matrix $\mathbf{A}$ was preprocessed by simply removing columns whose sum of elements resulted in the smallest RSSI values. This approach managed to make the optimization problem less ill-conditioned, as the condition number of matrix $\mathbf{F}_1$ improved to approximately 10^5 when the number of columns removed was optimum, i.e., generated the smallest coordinate or floor mean absolute error (MAE) during the validation phase. Fig. 3 illustrates MAE for longitude and latitude as the number of columns removed increases. The WAPs corresponding to the smallest RSSI sum were eliminated first.

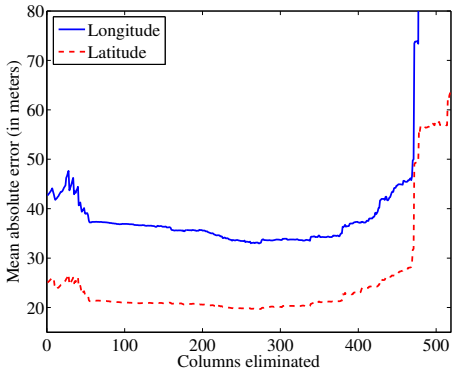

Fig. 3. MAE for longitude and latitude with respect to number of columns removed achieved by the CG algorithm using the first database.

TABLE I presents the MAE values for longitude, latitude and floor for the best configuration. The CG algo-

rithm achieved 74.26% correct floor identification and 24.75% missed by error of one floor.

TABLE I
RESULTS FOR THE FIRST DATA SET USING CG

	Coordinates (in meters)		Floor
	Longitude	Latitude	-
MAE	32.960	19.713	0.2601
Columns eliminated	273	274	248

For the second data set, equation (5) led to a matrix $\mathbf{F}_2 \in \mathbb{R}^{136\times136}$ with condition number 2.7338×10^7. In Fig. 4, as the columns were eliminated, the MAE only got larger, differently from the results obtained for the first data set, for which column elimination provided an initial improvement of the results. The order of elimination of columns was the same used for the first data set: columns with the smallest $l_2-norms$ were eliminated first.

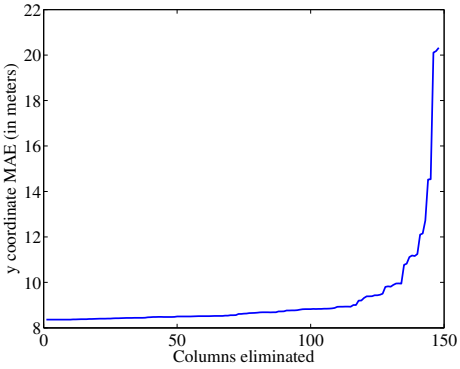

Fig. 4. MAE for coordinate y with respect to the number of columns removed using the second data set.

TABLE II shows the MAE for the floor and each coordinate for the second data set. In this case, the CG algorithm achieved 50.58% correct floor identification and 40.76% missed by error of one floor.

TABLE II
RESULTS FOR THE SECOND DATA SET USING CG

	Coordinates (in meters)			Floor
	x	y	z	-
MAE	19.484	8.361	2.513	0.646
Columns eliminated	0	0	0	0

C. The LASSO Algorithm

We solved the LASSO cost function for 100 different values of λ, gradually increasing its value at each iteration. Thus, the solution path of $\mathbf{x}$ starts as a good estimate of the OLS solution and shrinks its elements towards to zero as λ increases. The first components of $\mathbf{x}$ that will be set to zero are the ones that correspond to the columns of the measurement matrix $\mathbf{A}$ which do not contribute significantly to the solution, but not necessarily those with the smallest l_2-norm.

In order to evaluate performance, we used the MATLAB© function `lasso`. This implementation first centralizes matrix $\mathbf{A}$ and vector $\mathbf{b}$, i.e., subtracts each column by its mean. By doing so, it is necessary to apply exactly the same transformation to the validation data. This centralization is given by the following linear transformation

$$\mathbf{T} = \mathbf{I}_{M\times M} - \frac{1}{M}\mathbf{1}_{M\times M}. \tag{17}$$

Applying the transformation matrix $\mathbf{T}$ on the validation matrix $\mathbf{V}$ and on the validation vector $\mathbf{d}$, we have

$$\bar{\mathbf{V}} = \mathbf{TV} \text{ and } \bar{\mathbf{z}} = \mathbf{Tz}. \tag{18}$$

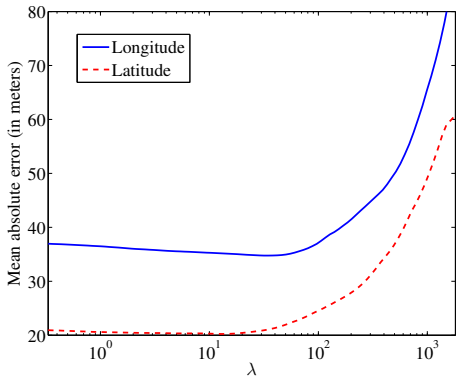

Fig. 5. MAE for latitude and longitude obtained by LASSO for the first data set.

The mean absolute error obtained by LASSO for the first database is presented in Fig. 5. In this case, the MAE first decreases, which means that the LS solution with all the columns does not yield the optimal result, for both latitude and longitude coordinates. This was also observed in the previous section for the CG algorithm, but here the LASSO algorithm itself eliminates selected variables and the associated columns of the measurement matrix. For a vector 157-sparse (a vector is n-sparse if it has n non-zero elements), the MAE obtained for latitude is 20.23 m, which is 2.6% larger than that obtained with the CG algorithm. For a vector 141-sparse, the MAE obtained for longitude is 34.76 m, which is 5.4% larger than that obtained with the CG algorithm. Although the MAE obtained from LASSO is slightly larger than that obtained with the CG method, using LASSO does not require the preprocessing stage. For the CG algorithm, harmful columns were removed manually to decrease the condition number of the measurement matrix. For LASSO, these harmful columns were selected and discarded automatically by the algorithm.

TABLE III
RESULTS FOR THE FIRST DATABASE USING LASSO

	Coordinates (in meters)		Floor
	Latitude	Longitude	-
MAE	20.23	34.76	0.23
Sparsity	157	141	78

Fig. 6 depicts the MAE for coordinate y using the second database. One can see that there is no minimum point beyond the first value of λ, meaning that, in contrast with the results obtained from the first data set, every column of measurement matrix interferes with the solution. TABLE IV show the MAE obtained for coordinates x, y, z and floor, which are practically equal to the results obtained via CG presented in TABLE II.

TABLE IV
RESULTS FOR THE SECOND DATA SET USING LASSO

	Coordinates (in meters)			Floor
	x	y	z	-
MAE	19.4	8.346	2.497	0.753
Sparsity	2	2	8	24

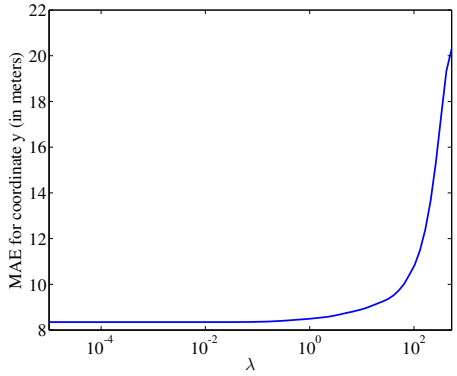

Fig. 6. MAE for coordinate y obtained by LASSO for the second data set.

Fig. 6 shows an interesting behavior of MAE as well: the error increases slowly. This behavior yields a more sparse vector and maintains the error within a bounded range. In TABLE V one can see that for a MAE only 5% higher than that provided by the OLS for coordinate y, the vector is 36-sparse. This result can be very useful to assist in the design of an infrastructure dedicated to indoor localization.

TABLE V

LASSO'S RESULTS FOR THE SECOND DATA SET FOR THE SOLUTION THAT REACHES MAE ONLY 5% HIGHER THAN THAT PROVIDED BY THE OLS ALGORITHM

	Coordinates (in meters)			**Floor**
	x	y	z	-
MAE	20.35	8.75	2.63	0.79
Sparsity	34	36	45	41

D. LASSO vs. Sparse Least Squares

For each value of λ used for solving equation (16), Fig.7 compares the solution obtained with LASSO and that derived from the OLS with the same sparsity level. The sparse OLS (sLS) estimator was obtained replacing by zero the coefficients of the OLS solution of equation (4) with the smallest absolute value. Fig. 7 shows the superior performance of LASSO and indicates that simply zeroing coefficients of the optimal OLS solution does not necessarily yield the best s-sparse solution.

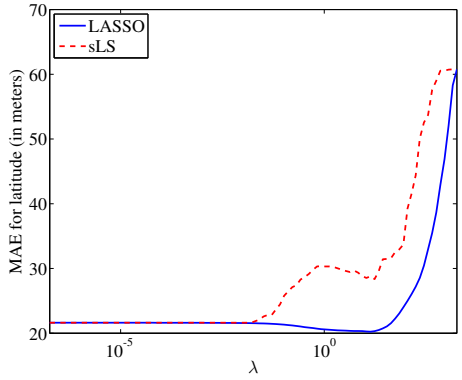

Fig. 7. MAE for latitude obtained by LASSO and sLS for the first data set.

VI. CONCLUDING REMARKS

In this work, we have shown two optimization approaches for the fingerprinting indoor localization problem: the first using conjugate gradient algorithm and the second using the LASSO algorithm. Simulations with two different data sets show that CG algorithm reaches a very low MAE, but when the data provides ill-conditioned problems, the CG method needs an exhaustive preprocessing stage. Even though the LASSO algorithm provides a slightly larger MAE, this approach has the advantage of skipping the preprocessing stage, due to its ability to eliminate components that seldom provide information to the problem. Furthermore, with LASSO it is possible to analyze the MAE produced by a determined sparsity number and verify which WAP is required to improve the localization. This result is very useful assisting in the project of an infrastructure dedicated to fingerprinting indoor localization, saving significant amount of effort and resources.

ACKNOWLEDGMENTS

This work was partially funded by CNPq, CAPES (Prodefesa 23038.009094/2013-83), and FINEP (Comunicações Submarinas FINEP-01.13.0421.00). The authors wish to thank Prof. Rafael S. Campos for sharing the data set of measurements taken at the State University of Rio de Janeiro, and Mr. Igor M. Quintanilha for invaluable help with database scripts.

REFERENCES

[1] Y. Wang, Q. Ye, J. Cheng, and L. Wang, "RSSI-Based Bluetooth Indoor Localization," *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp. 165–171, 2015.

[2] D. Arnitz, U. Muehlmann, and K. Witrisal, "Characterization and modeling of UHF RFID channels for ranging and localization," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 5, pp. 2491–2501, 2012.

[3] Z. Belhadi and L. Fergani, "Fingerprinting methods for RFID tag indoor localization," *International Conference on Multimedia Computing and Systems -Proceedings*, vol. 0, pp. 717–722, 2014.

[4] H.-S. Kim and J.-S. Choi, "Advanced indoor localization using ultrasonic sensor and digital compass," *2008 International Conference on Control, Automation and Systems*, pp. 223–226, 2008.

[5] R. A. F. Cunha, V. M. D. Pinho, and M. L. R. Campos, "O Método do Gradiente Conjugado Aplicado à Localização em Ambientes Fechados," *Anais do XXXIII Simpósio Brasileiro de Telecomunicações*, 2015.

[6] T. Chuenurajit, S. Phimmasean, and P. Cherntanomwong, "Robustness of 3D indoor localization based on fingerprint technique in wireless sensor networks," *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 1–6, 2013.

[7] G. Ding, Z. Tan, J. Zhang, and L. Zhang, "Fingerprinting localization based on affinity propagation clustering and artificial neural networks," *IEEE Wireless Communications and Networking Conference, WCNC*, pp. 2317–2322, 2013.

[8] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," pp. 409–436, 1952.

[9] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[10] A. Antoniou and W.-S. Lu, *Practical Optimization*. Boston, MA: Springer US, 2007.

[11] A. N. Tikhonov, "Solution of Incorrectly Formulated Problems And Regularization Method," *Doklady Akademii Nauk SSSR, 151*, vol. 151, no. 3, pp. 501–504, 1963.

[12] R. Tibshirani, "Regression shrinkage and selection via the lasso: a retrospective," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 3, pp. 273–282, 1996.

[13] J. Torres-Sospedra, R. Montoliu, A. Martinez-Uso, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," *IPIN 2014 - 2014 International Conference on Indoor Positioning and Indoor Navigation*, no. October, pp. 261–270, 2015.

[14] A. G. Souza, A. V. Marçal, and I. K. Ferreira, "Levantamento de Redes Wi-Fi 802.11 no Campus da UERJ." *Final Project (in portuguese)*, p. 122, 2008.