Oussar et al. EURASIP Journal on Wireless Communications and Networking 2011, 2011:81
http://jwcn.eurasipjournals.com/content/2011/1/81

EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Indoor localization based on cellular telephony RSSI fingerprints containing very large numbers of carriers

Yacine Oussar[1], Iness Ahriz[1], Bruce Denby[1,2*] and Gérard Dreyfus[1]

Abstract

A new approach to indoor localization is presented, based upon the use of Received Signal Strength (RSS) fingerprints containing data from very large numbers of cellular base stations–up to the entire GSM band of over 500 channels. Machine learning techniques are employed to extract good quality location information from these high-dimensionality input vectors. Experimental results in a domestic and an office setting are presented, in which data were accumulated over a 1-month period in order to assure time robustness. Room-level classification efficiencies approaching 100% were obtained, using Support Vector Machines in *one-versus-one* and *one-versus-all* configurations. Promising results using semi-supervised learning techniques, in which only a fraction of the training data is required to have a room label, are also presented. While indoor RSS localization using WiFi, as well as some rather mediocre results with low-carrier count GSM fingerprints, have been discussed elsewhere, this is to our knowledge the first study to demonstrate that *good quality* indoor localization information can be obtained, in diverse settings, by applying a machine learning strategy to RSS vectors *that contain the entire GSM band*.

1. Introduction

The accurate localization of persons or objects, both indoors and out of doors, is an interesting scientific challenge with numerous practical applications [1]. With the advent of inexpensive, implantable GPS receivers, it is tempting to suppose that the localization problem is today solved. Such receivers, however, require a minimum number of satellites in visibility in order to function properly, and as a result become virtually unusable in 'urban-canyon' and indoor scenarios.

The use of received signal strength measurements, or RSS, from local beacons, such as those found in Wi-Fi, Bluetooth, Infrared, or other types of wireless networks, has been widely studied as an alternative solution when GPS is not available [2-9]. A major drawback of this approach, of course, is the necessity of installing and maintaining the wireless networking equipment upon which the system is based.

Solutions exploiting RSS measurements from radiotelephone networks such as GSM and CDMA, both for indoor and outdoor localization, have also been discussed in the literature [10-14]. The near-ubiquity of cellular telephone networks allows in this case to imagine systems for which the required network infrastructure and maintenance are assured from the start, and recent experimental results [15-17] have furthermore suggested that efficient indoor localization may be achievable in a home environment using RSS measurements in the GSM band. The main contribution of the present article is to demonstrate conclusively that GSM can indeed provide an attractive alternative to WiFi-based and other techniques for indoor localization, *as long as the GSM RSS vectors used are allowed to include the entire GSM band*. The article outlines a new technique for accurate indoor localization based on RSS vectors containing up to the full complement of more than 500 GSM channels, derived from month-long data runs taken in two different geographical locations.

Input RSS vectors of such high dimensionality are known to be problematical for simple classification and regression methods. In the present article, we analyze the RSS vectors with machine learning tools [18,19] in order to extract localization information of good quality. The use of statistical learning techniques to analyze real

* Correspondence: denby@ieee.org
[1]Signal Processing and Machine Learning Laboratory, ESPCI - ParisTech, 10 rue Vauquelin, 75005 Paris, France
Full list of author information is available at the end of the article

or simulated WLAN and GSM RSS vectors has been discussed in [5,6,12], with promising results, however never using very high RSS dimensionalities such as those treated here. A second major contribution of our article is thus to demonstrate that good indoor localization can be obtained by extending machine learning-based localization techniques to RSS vectors of very high dimensionality, in this case the full GSM band. This use of the entire available set of GSM carriers–which may include base stations far away from the mobile to be located–allows the algorithms to extract a maximum of information from the radio environment, and thereby provide better localization than what is possible using the more standard approach of RSS vectors containing a few tens, at most, of the most powerful carriers.

It is worth stating from the outset that a classification approach to localization has been chosen in this work. In the literature examples may be found of localization treated as a problem of regression, i.e., estimating an actual physical position and quoting a mean positioning error ([3,12], etc.) or of classification, in which localization space is partitioned and the performance evaluated as a percentage of correct localizations ([4,11,15], etc.). One of the objectives of our research is to determine if measurements taken in different rooms can be grouped together reliably, which would allow to envisage, for example, a person-tracking system for use in a multi-room interior environment. It is for this reason that a classification approach was chosen here. This choice constitutes a third particularity of the approach presented in our article.

Section 2 of the article describes the experimental conditions and geographical sites at which the data were taken; the different RSS vectors used, which, following standard nomenclature, we call *fingerprints*, are also defined here. The machine learning techniques used are presented in Section 3, where we adopt a classification approach which labels each fingerprint with the index number of the room in which it was recorded. In Section 4, we introduce the idea of applying semi-supervised learning techniques to our datasets, in order to make our method applicable in the case where only a fraction of the training data are position-labeled. The semi-supervised approach is interesting, as has been pointed out, for example, in [4], because obtaining position labels for all points in a large dataset is expensive and time consuming. Finally, in Section 5, we present some conclusions and ideas for further study. An appendix provides basic information on the machine learning techniques used in the present investigation.

2. Measurement sites and datasets

2.1. Data-taking environment

The data used in our study were obtained by scanning the entire GSM band, which is one of the original aspects of our work. Two distinct datasets were created.

The first set, which we shall call the *home* set, was obtained using a TEMS GSM trace mobile [20], which is capable of recording network activity in real time and performing other special functions such as frequency scanning. Data were taken in a residence on the 5th (and top) floor of an apartment building in the 13th *arrondissement* of Paris, France. During the month of July, 2006, two scans per day were recorded in each of 5 rooms and manually labeled with a room number (1 to 5 as shown in Figure 1), yielding 241 full-GSM band scans, or about 48 scans per class. Scans must be initiated manually, and take about 2 min to complete. Each scan contained RSS and Base Station Identity Code information, BSIC, for each of 498 GSM channels, and occupies only a few kilobytes of data storage. Scans could be made at any point within a room; however, in practice, they were carried out in a subset of locations where the scanning device and laptop computer could be conveniently placed: tabletop, chair, etc. The exact positions of the individual scans were not recorded, which is consistent with the adopted classification approach to localization.

The second dataset, which we call here the *lab* set, was acquired with a different apparatus, a machine-to-machine, or M2M, GSM/GPRS module [21], which can be driven using standard and manufacturer-specific AT modem commands. Datasets were recorded on the second floor (beneath a wooden attic and a steel-sheet roof) of a

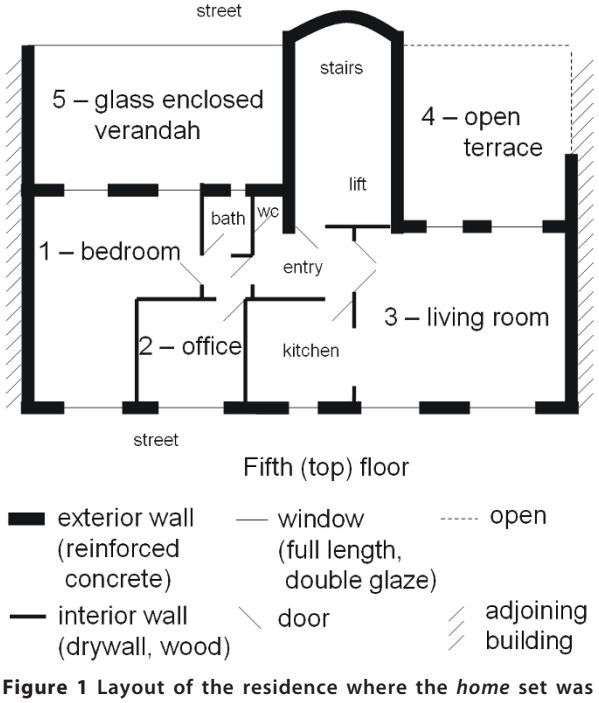

Figure 1 Layout of the residence where the *home* set was recorded.

research laboratory in the 5th *arrondissement* of Paris, France. A total of 600 GSM scans were carried out during the month of September, 2008, in five of the rooms of the laboratory, as indicated in Figure 2. Each *lab* set scan contains data from 534 GSM channels. This is more than in the *home* set since the somewhat older TEMS module used did not cover a portion of the band known as 'extended-GSM'. As in the home set, scans were labeled manually with a room number, and were recorded at positions where the measuring device could be easily placed. In contrast to the *home* set, in order to minimize interference with daily laboratory activities, the measurement device was always placed at nearly the same position in each room, as indicated by the stars in Figure 2.

For each dataset (*home* and *lab*), the identical measuring device (TEMS for *home*, M2M for *lab*) was used for all scans. Indeed, tests showed that training with one M2M device and testing on another often gave poor results. This effect was later found to be due to variations in the device antennas used, and could be eliminated in future work. Nevertheless, the use of two different types of devices for our data recording (TEMS and M2M), as well as the choice of acquisition sites which are well separated both geographically and in time, gives an indication of the general applicability of our method.

The TEMS trace mobile is in appearance identical to a standard GSM telephone, the trace characteristics being implemented via hardware modification to the handset. The M2M modems are essentially bare GSM modem chipsets meant to be incorporated into various OEM (original equipment manufacturer) products such as vending machines, vehicles, etc.

To give an idea of the behavior of GSM RSS values in an indoor scenario, Figure 3 shows the mean value of RSS of channels in two different rooms of the *Lab* set. It can be seen that RSS values at a given frequency are in general different for the two rooms. The classification algorithms exploit these differences.

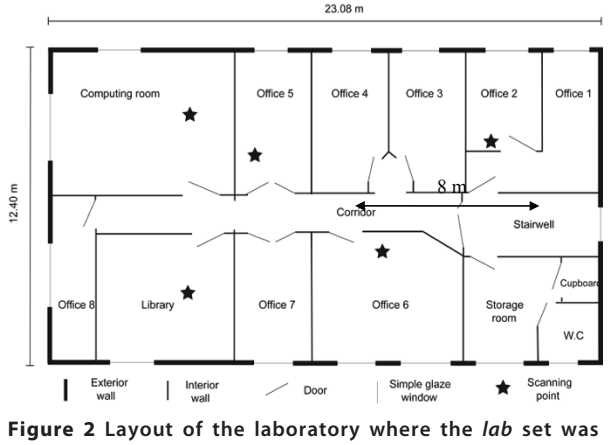

Figure 2 Layout of the laboratory where the *lab* set was recorded.

Most commercial implementations of fingerprint-based outdoor GSM localization exploit the standard Network Measurement Reports, NMR, which, according to the GSM norm, the mobile station transmits to its serving Base Transceiver Station (BTS) roughly twice per second during a communication. Each 7-element NMR contains the RSS measurements of fixed-power beacon signals emanating from the serving BTS and its six strongest neighbors. In contrast, the frequency scans recorded by our TEMS and M2M modules are performed in *idle* mode, that is, when no call is in progress. Although NMRs are thus not available in our data, the scans nonetheless contain data on *all* channels, and include, at least in principle, the BSIC of each channel. This allows, for example, to 'construct' an NMR artificially, as was done in the definition of the *Current Top 7* fingerprint in Section 2.2.

During a scan, in addition to obtaining the RSS value at each frequency, the trace mobile attempts to synchronize with the beacon signal in order to read the BSIC value. Failure to obtain a BSIC can occur for two reasons: (1) the signal to noise + interference ratio is poor, perhaps because the BTS in question is located far from the mobile; or (2) the channel being measured is a *traffic* channel which therefore does not contain a BSIC. As traffic channels are not emitted at constant power and may employ frequency hopping, one might initially conclude that they will not useful for localization (as the hopping sequence is unknown, an RSS value in this case just represents the observed power at a given frequency, averaged over a few GSM frames). Rather than introduce this bias into our data *a priori*, we chose to ignore BSICs and allow the variable selection procedure to decide which inputs were useful. This choice is not without cost, as it does not guarantee that from one scan to the next the data at a particular frequency is always from the same BTS. As we shall discover later, however, traffic channels do in fact turn out to be amongst those selected by the learning algorithm as being important.

As described earlier, to create a database entry, a human operator manually positions the trace mobile, initiates the scan, and labels the resulting RSS vector with its class index (i.e., room number). The training set thus accumulated over a period of time can then be used to build a classifier capable of labeling new RSS vectors obtained in the same geographical area. In such a supervised training scenario, the necessity of an extensive hand-labeled training set for each measurement site is clearly a drawback. For this reason we also examine, in Section 4, semi-supervised training techniques, which require only a fraction of the database entries to be labeled.

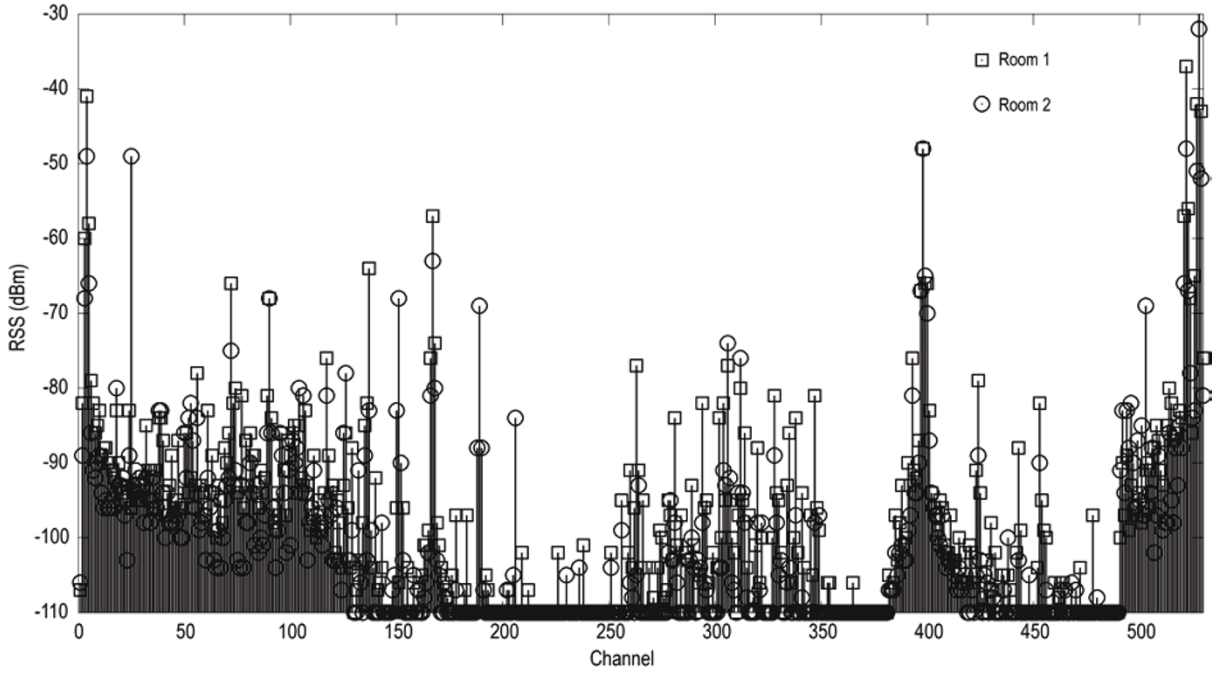

Figure 3 RSS scans performed in two different rooms (*Lab* set).

2.2. Preprocessing and variable selection

In the *home* (TEMS) scans, 10 empty carrier slots which always contained a small, fixed value were removed, leaving 488 values. This procedure was not found necessary for the *lab* (M2M) scans, and all 534 carriers were retained. For both scan sets, the total number of dataset entries is quite limited compared to the dimensionality of the RSS vectors. To address this problem, three types of fingerprints, containing subsets of carriers, were defined as described below.

In the following, we denote by $N_{\max}$ the total number of carriers in the carrier set under study: $N_{\max}$ = 488 in the *home* scans, and $N_{\max}$ = 534 for the *lab* scans. We define matrix **RSS** as the *full observation matrix*, whose element RSS_{ij} is the strength value of carrier j in dataset entry i. In other words, each row of **RSS** contains the received signal strength values measured at a given location, and each column contains the received signal strength values of a given carrier in the carrier set under investigation. Thus, **RSS** has M rows and $N_{\max}$ columns, where M is the number of dataset entries (i.e., the number of GSM band scans in the dataset).

All N_{max} Carriers

This fingerprint includes the entire set of carriers, i.e., each column of RSS is a fingerprint, of dimension $N_{\max}$. Its consequent high dimensionality limits the complexity of the classifiers which can be used in its evaluation, as we shall see in the presentation of the results.

N Strongest

The *N Strongest* fingerprint contains the RSS values of the N carriers which are strongest when averaged over the entire training set. Therefore, it involves a *reduced observation matrix* $\mathbf{RSS}_1$, derived from the full observation matrix by deleting the columns corresponding to carriers that are not among the N strongest on the average; therefore, $\mathbf{RSS}_1$ has M rows and N columns. The value of N is determined as follows: the strongest (on average) carrier is selected, a classifier is trained with this one-dimensional fingerprint, and the number of correctly classified examples on the validation set (see Section 3.1 on model training and selection) is computed. Another classifier is trained with the (two-dimensional) fingerprint comprised of the measured RSS values of the strongest and second strongest carriers. The procedure is iterated, increasing the fingerprint dimension by appending successively new carriers, in order of decreasing average strength, to the fingerprint. The procedure is stopped when the number of correctly classified examples of the validation set no longer increases significantly. N is thus the number of carriers which maximizes classifier performance. It may be different for different types of classifiers, as shown in the results section; it is typically in the 200-400 range.

Current Top 7

As mentioned earlier, since our scans were obtained in idle mode, we do not have access to standard NMRs.

It is nevertheless interesting to have a 'benchmark' fingerprint of low dimensionality to which we may compare results obtained with our 'wider' fingerprints. This is the role of *Current Top 7*. While it would be desirable to use as fingerprint of location i the vector of measured strengths of the seven strongest carriers *at location i*, this is problematical since most classifiers require an input vector of fixed format. Therefore, the *Current Top 7* fingerprint is defined as follows: it contains the measured strengths of the carriers which were among the seven strongest on at least one training set entry. This fingerprint has a fixed format, for a given training set, and a typical length of about 40 carriers for our data. Therefore, in this context, the reduced observation matrix $\mathbf{RSS}_2$ has M rows and about 40 columns. In each row, i.e., for a given GSM band scan, only seven elements are defined; the remaining elements of the row are simply set to zero.

Once a fingerprint has been chosen, a subsequent principal component analysis (PCA, see appendix) can be applied in order to obtain a further reduction in dimensionality. This allows us to construct more parsimonious classifiers, which can then be compared to those which use the primary variables only.

3. Supervised classification algorithms

An introduction to supervised classification by machine learning methods is provided in the Appendix, with emphasis on the classification method (support vector machines) and preprocessing technique (principal component analysis) adopted in the present article.

3.1. Model training and selection

We consider the indoor localization problem as a multiclass classification problem, where each room is a class. Therefore, given a fingerprint that is not present in the training dataset, the classifier should provide the label of the room where it was measured. We describe in Section 3.2 two strategies that turn multiclass classification problems into a combination of two-class (also termed 'binary' or 'pairwise') classification problems; therefore, the present section focuses on training and model selection for two-class classifiers.

Since the size of the training set is not very large with respect to the number of variables, support vector machine classifiers were deemed appropriate because of their built-in regularization mechanism. For each classification problem, the Ho-Kashyap algorithm [22] was first run in order to assess the linear separability of the training examples. Linear support vector machines were implemented whenever the examples turned out to be linearly separable. Otherwise, a Gaussian kernel support vector machines (SVM) was implemented:

$$K(x, \mathbf{y}) = \exp \frac{\|\mathbf{x} - \mathbf{y}\|^2}{\sigma^2} \tag{1}$$

where σ is a hyperparameter whose value is obtained by cross-validation (see below).

As usual, a GSM environment described by the fingerprint $\mathbf{x}$ is classified according to the sign of

$$f(x) = \sum_{i=1}^{M} \alpha_i y_i (x_i, x) + b \tag{2}$$

where α_i and b are the parameters of the classifier, $y_i = \pm 1$ and $\mathbf{x}_i$ are the class label and the fingerprint of dataset entry i (i.e., row i of $\mathbf{RSS}$, $\mathbf{RSS}_1$, or $\mathbf{RSS}_2$ depending on the fingerprint used by the classifier), respectively, and $K(.)$ is the chosen kernel.

The values of the width σ of the kernel, and of the regularization constant (see appendix), were determined by cross-validation (CV), and the performance of the selected models were subsequently assessed on a separate test set, consisting of 20% of the available dataset. Six-fold CV was performed on the remaining data for the *home* set, and 10-fold CV for the larger *lab* set. In order to assess the variability of the cross-validation score with respect to data partitioning, each CV procedure was iterated ten times with random shuffling of the database entries before each iteration. As a result, a mean CV score was computed along with an estimate of its standard deviation. The test set, throughout, always remains the same. The overall procedure is illustrated diagrammatically in Figure 4, for six-fold cross-validation.

As the procedure outlined corresponds to supervised classification, all dataset entries are labeled. The numbers of examples of each class were balanced in each fold.

The SVMs used in our study, both with linear and Gaussian kernels, were implemented using the Spider toolbox [23].

In order to obtain baseline results, K-nearest neighbor (K-NN) classifiers using the Euclidean distance in RSS-space were implemented. The hyperparameter K was determined by the same cross-validation procedure as for SVM's.

3.2. Decision rules for multiclass discrimination

When the discrimination problem involves more than two classes, it is necessary, for pairwise classifiers such as SVM, to define a method that allows to combine multiple pairwise classifiers into a single multiclass classifier. This can be done in two ways: one-vs-all and one-vs-one.

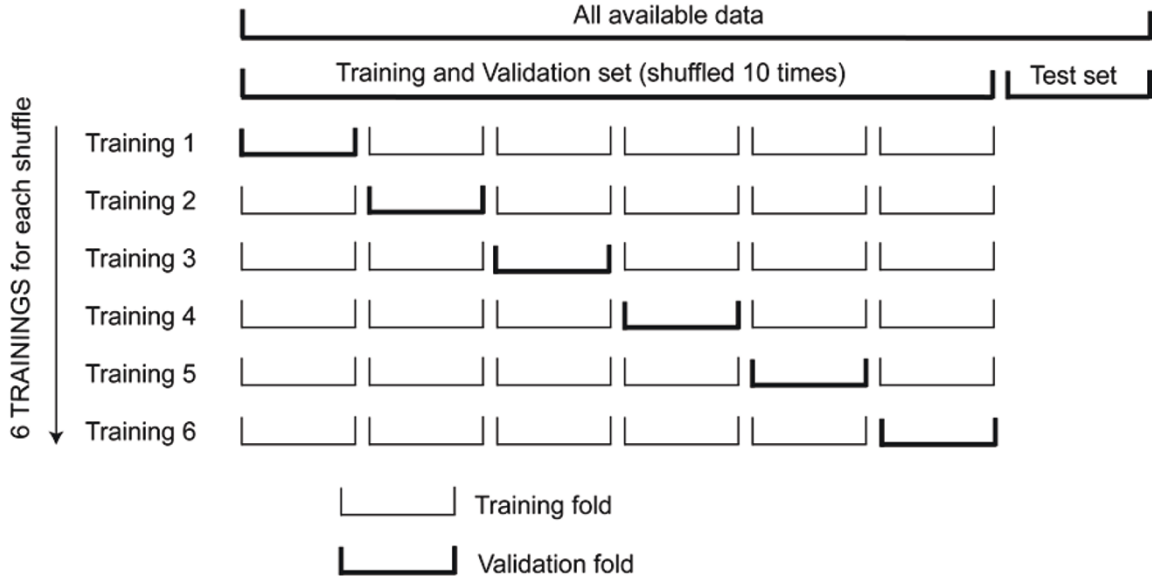

Figure 4 Partition of the data into folds for the cross-validation procedure.

3.2.1. The one-vs-all approach

The one-vs-all approach consists of dividing the multiclass problem into an ensemble of pairwise classification problems. Thus, for a problem with n classes, the resulting architecture will be composed of n binary classifiers, each specialized in separating one class from all the remaining ones. Figure 5 illustrates the procedure. Each of the n classifiers is trained separately, whereas validation is carried out using the architecture indicated in the figure. To localize a test set example, the outputs of all n classifiers are first calculated; following the conventional procedure, the predicted class is taken to be that of the classifier with the largest value of $f(\mathbf{x})$ (relation (2)). The one-vs-all technique is advantageous from a computational standpoint, in that it only requires a number of classifiers equal to the number of classes, in our case, 5.

3.2.2. One-vs-one classification

This approach decomposes the multiclass problem into the set of all possible one-vs-one problems. Thus, for an n-class problem, $\frac{n(n-1)}{2}$ classifiers must be designed. Figure 6 illustrates the architecture associated with this method.

The decision rule in this case is based on a vote. First, the outputs of all classifiers are calculated. Now let $C_{i,j}$ be the output of the classifier specializing in separating class i from class j. If $C_{i,j}$ is 1, the tally for class i is

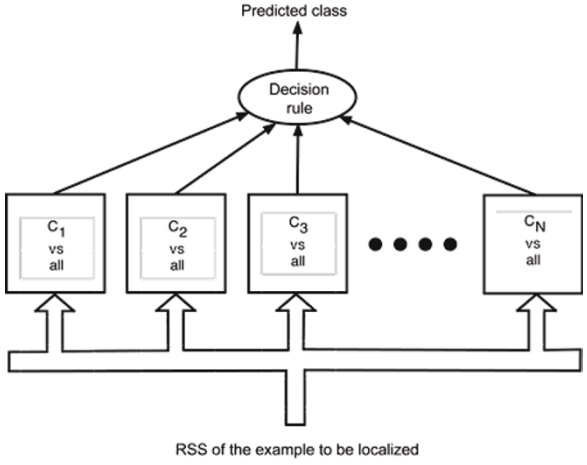

Figure 5 One-vs-all classification.

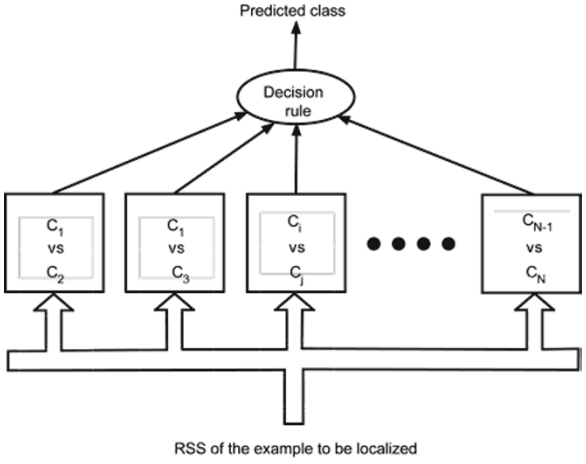

Figure 6 One-vs-one classification.

incremented; if it is -1, the class tally of class j is increased by 1. Finally, the class assigned to the example is that having the highest vote tally.

A disadvantage of the one-vs-one technique is of course the increase in the number of classifiers required as compared to one-vs-all. In our case of five classes, 10 classifiers are required, which still remains manageable.

3.3. Results

In order to assess the accuracy and robustness of our approach, results are presented on datasets which have been:

- recorded at two different locations;
- taken at moments widely separated in time (approx. 2 years);
- realized under substantially different experimental conditions.

The performance of each classifier is presented as the percentage of test set examples which are correctly classified. There is no rejection class.

On the *home* set, when PCA is not used, the number of input variables exceeds the number of training set examples for all but the *Current Top 7* fingerprint. Using geometrical arguments, Cover's theorem [24] states that in this case, the training set will always be linearly separable, which can of course also be verified using the Ho-Kashyap algorithm. From a practical standpoint, this means that, due to the small size of the training set, it is not meaningful to test non-linear classifiers on these fingerprints (unless a dimensionality reducing PCA is applied first).

This difficulty is less frequently posed in the lab set, which is of somewhat larger size. Cover's theorem in fact comes into play here only in the cases of one-vs-one classifiers (with the exception of the Current Top 7 fingerprint), and of one-vs-all classifiers applied to the *All* $N_{\max}$ *Carriers* fingerprint.

3.3.1. Results on the home set

We recall that the *home* set is composed of 241 scans containing RSS vectors with 488 GSM carriers. Of the 241, 61 scans are chosen at random to make up the test set. The remaining 180 examples are used to tune and select classifiers using the cross validation strategy.

Table 1 presents the classification results for SVMs with linear and Gaussian kernels, respectively (see Section 3.1) in one-vs-one and one-vs-all configurations. Results for a K-NN classifier, without PCA, are also given, for comparison. It was found unnecessary to test the Gaussian SVM on the one-vs-one scenario, as the application of the Ho-Kashyap algorithm revealed that the training sets were always linearly separable in this case; the corresponding entries are indicated with an asterisk. Similarly, the Ho-Kashyap algorithm showed that training sets were not linearly separable in the case of one-vs-all classifiers with PCA: as expected, nonlinear SVM classifiers perform better than linear ones in that case. Finally, it is not meaningful to apply the Gaussian SVM to the *All* $N_{\max}$ *Carriers* fingerprint, due to Cover's theorem; this entry is indicated with a double asterisk. Wherever PCA was used in the table, the optimal number of principal components is indicated in parentheses, as is the optimal value of K for the K-NN classifier.

From Table 1, we may immediately remark that the *Current Top 7* fingerprint, which is meant to mimic a

Table 1 Percentage of correctly classified test set examples (*home* set)

Classifier	Current Top 7	*N* Strongest	All N_{max} (= 488) carriers
		Linear SVM	
One-vs-one			
w/PCA	57.4 (*PC* = 8)	96.7 (*N* = 360, *PC* = 8)	96.7 (*PC* = 8)
w/o PCA	68.9	95.1 (*N* = 210)	96.7
One-vs-all			
w/PCA	62.3 (*PC* = 8)	85.2 (*N* = 420, *PC* = 4)	85.2 (*PC* = 4)
w/o PCA	60.6	98.4 (*N* = 340)	95.1
		Gaussian SVM	
One-vs-one	*	*	*
One-vs-all			
w/PCA	65.6 (*PC* = 8)	*88.5* (*N* = 420, *PC* = 4)	88.5 (*PC* = 4)
w/o PCA	68.8	98.4 (*N* = 140)	**
K-NN	54.1 (*K* = 7)	95.1 (*N* = 240, *K* = *10*)	91.8 (*K* = 12)

N is the number of carriers used in *N Strongest*. The optimal number of principal components *PC*, and optimal *K* of the *K*-NN classifier, are given in parentheses.
*It was unnecessary to apply the Gaussian SVM to the one-vs-one case because the training sets were always found to be linearly separable using Ho-Kashyap.
**It is not meaningful to apply the Gaussian SVM to the *All* $N_{\max}$ *Carriers* fingerprint, due to Cover's theorem (see text).

standard 7-carrier NMR, never provides better than 69% classification efficiency. In comparison, when the RSS vectors are extended to include the strongest 340 carriers, for example, a linear, one-vs-all SVM correctly classifies 98.4% of the test set examples. Indeed, when large numbers of carriers are retained, seven of the nine SVM classifiers presented in the table are able to correctly classify over 95% of the test set examples. The application of PCA to the high carrier count fingerprints leads to a performance degradation in the one-vs-all mode, which can be recovered, however, by preferring the more sensitive one-vs-one approach. The principal result, which including large numbers of GSM carriers in the RSS fingerprints leads to very good performance, is very clear.

3.3.2. Results on the lab set

The *lab* dataset is made up of 601 scans containing RSS vectors of 534 carriers. A test set was constructed from 101 randomly selected scans, leaving 500 for the cross-validation procedure.

Table 2 shows the classification results for linear and Gaussian SVMs in the one-vs-one and one-vs-all configurations, with results from a non-PCA *K*-NN classifier also provided for comparison. The meaning of the asterisk entries is the same as in Table 1. The results on the *lab* set exhibit many similarities to those on the *home* set. First, it is once again clear that the NMR-like *Current Top 7* fingerprint is inadequate for providing good localization performance; indeed, its performance here is even worse than that on the *home* set. Secondly, we note that very good performance can be obtained by extending the fingerprint to a much larger number of carriers. For example, a linear one-vs-all SVM acting upon a fingerprint of the strongest 390 carriers here correctly classifies 95.1% of the test set examples. Finally, the application of PCA in the one-vs-all case again leads to a degradation in performance. In contrast to the *home* set, however, this degradation is not recoverable here by using a one-vs-one classifier. Indeed, the classification problem appears to be globally more difficult for the *lab* set than for the *home* set, as is further evidenced by the fact that only four of the nine high carrier count SVM classifiers obtain more than 95% correct identification, compared to seven out of nine for the *home* set. The performance of the *K*-NN classifier is also substantially lower than on the *home* set, and, as already mentioned, the overall performance of the Current Top 7 fingerprint on the *lab* set is very poor. The best result on the *lab* set, however, is 100% correct identification on the independent test set, verifying once again that good localization performance can indeed be obtained by applying machine learning techniques to fingerprints with large numbers of carriers. Based on the size of the rooms involved, this localization performance corresponds to a positional accuracy of some 3 m. As in the case of the *home* set, one-vs-all linear classifiers with PCA perform poorly.

4. Semi-supervised classification

As was pointed out earlier, the RSS scans are manually labeled during data acquisition. In large-scale environments, this is a tedious and time consuming task, which impinges in a negative way on the future development of real world applications of the localization techniques proposed here. A more favorable scenario would be one in which the acquisitions take place automatically, and the user is required to intervene only occasionally to provide labels to help the learning algorithm discover

Table 2 Percentage of correctly classified test set examples (lab set)

Classifier	Current Top 7	*N* Strongest	All N_{max} (= 534) carriers
		Linear SVM	
One-vs-one			
w/PCA	38.6 (*PC* = 8)	70.3 (*N* = 490, *PC* = 10)	70.3 (*PC* = 8)
w/o PCA	35.6	98 (*N* = 280)	100
One-vs-all			
w/PCA	32.6 (*PC* = 8)	59.6 (*N* = 520, *PC* = 10)	59.6 (*PC* = 10)
w/o PCA	45.5	95.1 (*N* = 390)	94.1
		Gaussian SVM	
One-vs-one	*	*	*
One-vs-all			
w/PCA	49.5 (*PC* = 10)	76.6 (*N* = 530, *PC* = 10)	68.3 (*PC* = 10)
w/o PCA	54.5	96.6 (*N* = 290)	**
K-NN	52.5 (*K* = 6)	68.3 (*N* = 320, *K* = 13)	71.3 (*K* = 10)

N is the number of carriers used in *N Strongest*. The optimal number of principal components *PC*, and optimal *K* of the *K*-NN classifier, are given in parentheses.
*It was unnecessary to apply the Gaussian SVM to the one-vs-one case because the training sets were always found to be linearly separable using Ho-Kashyap.
**It is not meaningful to apply the Gaussian SVM to the *All Nmax Carriers* fingerprint, due to Cover's theorem (see text).

the appropriate classes. Semi-supervised learning algorithms function in exactly this way.

Several methods of performing semi-supervised classification are described in the machine learning literature [25,26]. Encouraged by the good performance obtained with supervised SVMs, we have chosen to test a kernel-based semi-supervised approach known as the Transductive SVM, or TSVM [27], which has been applied with success, for example, in text recognition [27] and image processing [28].

A TSVM functions similarly to a standard SVM, that is, by finding the hyperplane which is as far as possible from the nearest training examples, with the key difference that some of the examples have class labels, and others do not. The TSVM learning algorithm consists of two stages:

- In the *first stage*, a standard SVM classification is performed using only the labeled data. The classification function of Equation 2 is then used to assign classes to the unlabeled points in the training set.
- The *second stage* of the algorithm solves an optimization problem whose goal is to move the unlabeled points away from the class boundary by minimizing a cost function. This function is composed of a regularization term and two error-penalization terms, one for the labeled examples, and the other for those which were initially unlabeled (and for which labels were predicted in the *first stage*). The optimization is carried out by successive permutation of the predicted labels. Permutations of two labels which lead to a reduction in the cost function are carried out, while all others are forbidden. The optimization terminates when no further permutations are possible.

As in the case of standard SVMs, regularization and the use of a nonlinear kernel introduce hyper-parameters whose values are to be estimated during the cross-validation process. In our study, the TSVM was implemented using the SVM^{light} toolbox [29].

The presence of unlabeled data renders a data partition like that of Figure 3 impossible. In order to build a classifier with the best possible generalization performance, we have defined a new partition which differs from the one traditionally proposed [27,30]. The procedure is described below.

A test set is first chosen at random from the labeled data. The remaining data are then divided into two subsets, one for the validation, and a second which is mixed with the unlabelled data to form a training set of partially labeled data. The principle is illustrated in Figure 7.

The results are presented in the next section. A *K*-NN classifier was also evaluated, for comparison. *K*-NN cannot make use of the unlabeled data: the nearest neighbors that are relevant for classifying an entry are its *labeled* neighbors only. The hyper-parameter *K* was determined in the validation procedure.

4.1. Results

We note first that since the class labels of many of the training examples are unknown, it is not possible to carry out a one-vs-one strategy. Thus, only the one-vs-all approach was implemented here.

4.1.1. Results on the home set

In order to make the performances of the TSVM classifiers directly comparable to those obtained using SVMs, the test set was chosen to be the same 61 example one that was used to make Table 1. The data partition was implemented as indicated in Figure 6, allocating 40 examples to the validation set, and 140 to the training set, 100 of which are unlabeled. This choice thus imitates a scenario in which some 80/180 = 44% of the data is labeled (where we consider that the test set is used here only for purposes of evaluating the viability of our method).

Table 3 presents the test set performances obtained, in percent, for the classifiers that were implemented. As was the case for the supervised classifiers, the *Current Top 7* fingerprint achieves only mediocre performance. For the classifiers which use large numbers of carriers,

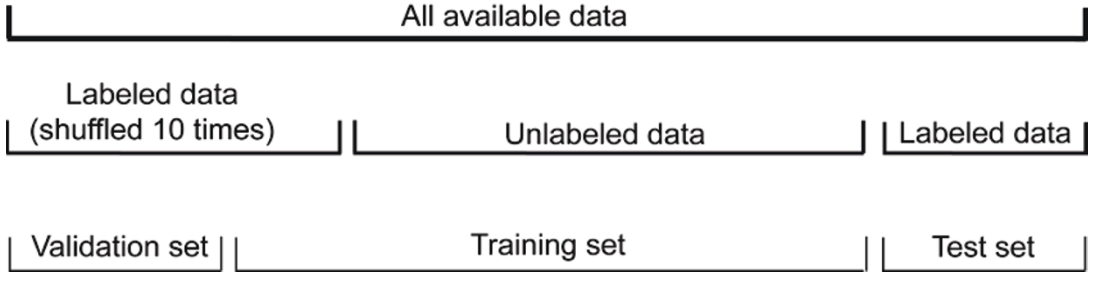

Figure 7 An original data partitioning scheme for semi-supervised learning.

Table 3 Percentage of correctly classified test set examples for the TSVM (*home* set)

TSVM Classifier	Current Top 7	*N* Strongest	All N_{max} (= 488) carriers
		Linear	
w/PCA	54.1 (*PC* = 4)	95.1 (*N* = 350, *PC* = 4)	93.4 (*PC* = 4)
w/o PCA	55,7	98.4 (*N* = 370)	98.4
		Gaussian	
w/PCA	52.5 (*PC=10*)	98.4 (*N* = 280, *PC* = 6)	96.7 (*PC* = 7)
w/o PCA	62,3	98.4 (*N* = 330)	-
K-NN	50.8 (*K=4*)	91.8 (*N* = 200, *K* = 4)	86.8 (*K* = 5)

The definitions of *K, N*, and *PC* are identical to those used in Tables 1 and 2.

however, seven of the eight tested were able to correctly classify over 95% of the test set examples. Furthermore, the performance of the linear TSVM classifier without PCA is identical to that obtained by the same type of classifier trained in supervised mode, thus demonstrating that semi-supervised learning techniques are indeed an interesting approach for the localization problem. Also, a simple linear classifier is apparently adequate here, as the Gaussian TSVM did not provide any improvement in performance. A *K*-NN classifier performs poorly in this case because of the small number of labeled examples in the training set.

4.1.2. Results on the lab set

We recall that the *lab* dataset contains 601 scans. The test set of 101 examples that was used to create Table 2 is again employed for the TSVM. The training set here contains 400 examples, of which 100 are labeled, with the validation being performed on the 100 remaining examples. Thus, for the *lab* set, the operating scenario is one in which 200/500 = 40% of the data is labeled, the 101 examples of the test being used only to evaluate the validity of our approach.

Table 4 summarizes the performances of the classifiers tested. As was the case for supervised learning case, the classification problem of the *lab* set appears to be more difficult than that of the *home* set. The performance of the *Current Top 7* fingerprint for all classifiers, and the performance of the *K*-NN classifiers for all fingerprints, are again poor. The best performance, 87.1% here, is again obtained with a linear TSVM and a fingerprint of 350 carriers without PCA, and is not improved when a non-linear TSVM is applied. The importance of including large numbers of carriers is once again demonstrated, even if the semi-supervised learning performance here, as compared to the fully supervised case, while good, is less impressive than on the *home* set.

5. Conclusion

We have presented a new approach to indoor localization, founded upon the inclusion of very large numbers of carriers in the GSM RSS fingerprints followed by an analysis with appropriate machine learning techniques. The method has been tested on datasets taken at two different geographical locations and widely separated in time. In both cases, room-level classification performance approaching 100% was obtained. To the best of our knowledge, this is the first demonstration that indoor localization of very good quality can be obtained from full-band GSM fingerprints, by making proper use of relatively unsophisticated machine learning tools. We have also presented promising results from a new variant of the TSVM semi-supervised machine learning algorithm, which should go a long way towards alleviating the difficulty of obtaining large numbers of position-labeled RSS fingerprints.

The results obtained in our study allow to imagine new localization services and applications which are of very low cost and complexity, due to being based upon the cellular telephone networks which today are almost ubiquitous throughout the world. In the study presented here, the localization algorithms were always executed

Table 4 Percentage of correctly classified test set examples (*lab* set)

TSVM Classifier	Current Top 7	*N* Strongest	All N_{max} (= 534) carriers
		Linear	
w/PCA	40.6 (*PC* = 10)	60.4 (*N* = 260, *PC* = 10)	62.4
w/o PCA	32.7	87.1 (*N* = 350)	81.2
		Gaussian	
w/PCA	38.6 (*PC* = 10)	47.5 (*N* = 250, *PC* = 10)	48.5
w/o PCA	37.6	75.2 (*N* = 350)	-
K-NN	37.6 (*K* = 6)	55.5 (*N* = 450, *K* = 5)	55.4 (*K* = 5)

offline on standard processors. In future, such a system could be implemented either on the handset or on a server. In the first case, the GSM band scan and location estimation calculations are performed in the handset itself; in the second, GSM band scans performed by the handset are sent to a server, where the position is estimated.

A more ambitious measurement campaign, including several more geographical locations, finer positioning grids, and multiple RSS measuring devices, is currently in the development stage. In addition to helping assess the viability of our approach over a wider range of environments, this study will also allow us to answer certain questions which were not addressed in the current work, for example:

- What is the ability of the method to identify on which the floor of a building a mobile is localized?
- How will the performance behave in environments with relatively poor GSM coverage (rural areas, etc.)?
- What is the true nature of the time stability of the method? Will the database need to be updated regularly and if so on what time scale? Although our tests showed that coherence over a one-month period is possible, these temporal aspects need to be evaluated rigorously.

Studies of additional types of semi-supervised learning algorithms, as well as methods of predicting RSS values, are envisioned in order to continue to address the time consuming labeling task in large scale environments. An exploration of time-dependent modeling techniques, a more elaborate variable selection procedure, a more sophisticated multiclass discrimination approach, and the incorporation of other types of sensors in our measuring devices for added redundancy, are also envisioned.

Appendix

We provide here basic information that may be useful to readers who are not familiar with supervised classification by statistical machine.

Supervised classification by machine learning

Supervised classification consists of assigning one class, out of several known classes, to an object described by a vector of variables (also termed 'descriptors') $\mathbf{x}$. In the present article, $\mathbf{x}$ is a GSM fingerprint. For simplicity, we consider here two-class problems: an object i belonging to class A has label $y_i = +1$, while an object belonging to class B has label $y_i = -1$; two extensions to multiclass problems are described in the text.

We take the traditional classifier design strategy that consists of (i) postulating a parameterized function $f(\mathbf{x}, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is a vector of adjustable parameters, and (ii) estimating the vector $\boldsymbol{\theta}$ such that the classification rule *the object described by* $\mathbf{x}$ *belongs to A if* $\text{sgn}(f(\mathbf{x},\boldsymbol{\theta})) > 0$, *and it belongs to B otherwise* classifies *all possible objects* of the two classes with a minimal rate of classification errors. The equation of the surface that separates the two classes in descriptor space is thus $f(\mathbf{x},\boldsymbol{\theta}) = 0$.

In order to estimate the parameters of the classifier, a database called *training set* is necessary; it contains a collection of objects ('examples') that are known and that have been labeled by a 'supervisor', hence the term 'supervised learning'. In the present study, fingerprint measurements have been performed, and each fingerprint has been recorded together with the label of the room where the measurement was performed. The difficulty of the training task stems from the fact that a *finite* number of examples are available, while the resulting classifier should be optimal for *all possible objects*: there is a risk that the classifier classify correctly all available examples but perform poorly on other objects of the class. Such a classifier is said to be *overfitted* to the training data; it *generalizes* poorly. Clearly, if the postulated function is given a very large number of adjustable parameters that can vary on an arbitrarily large scale, i.e., if the postulated function is very flexible, it may define a very complicated separation surface between the two classes, which classifies correctly all examples of the training set and generalizes poorly to other objects of the classes. One way to alleviate this problem consists of preventing the parameters from becoming too large; this is known as *regularization*. Conversely, if the postulated function is not complex enough, i.e., is too 'stiff', it may define a boundary surface that lacks flexibility to accommodate the training data, hence generalize poorly. Therefore, the central problem in classifier design by machine learning methods is that of finding a boundary surface of appropriate complexity; the complexity of a function is accurately defined by its *Vapnik-Cervonenkis* (*VC*) *dimension*, whose description goes beyond the scope of the present appendix.

Support vector machines

Support Vector Machines (SVMs) are classifiers that feature a built-in regularization mechanism, and are guaranteed to produce classifiers of optimal complexity.

First assume that the examples present in the training set are *linearly separable*, i.e., a postulated function of the form $f(\mathbf{x},\boldsymbol{\theta}) = \mathbf{x}\ \boldsymbol{\theta}$ provides a boundary surface that classifies all examples of the training set without errors. In other words, all examples of the training set can be perfectly separated by a straight line if $\mathbf{x}$ is of dimension 2, by a plane if $\mathbf{x}$ is of dimension 3, and by a hyperplane if descriptor space is of dimension larger than 3. Then

the parameters of the optimal hyperplane are obtained by solving numerically the following constrained optimization problem: minimize $||\boldsymbol{\theta}||$ under the constraint that all examples are correctly classified; these constraints are linear inequalities. The fact that $||\boldsymbol{\theta}||$ is minimized during training provides SVMs with an automatic regularization mechanism.

This constrained optimization problem can be expressed equivalently in a *dual form* by searching a solution under the form

$$\theta = \sum_{i=1}^{N} \alpha_i \gamma_i \mathbf{x}_i$$

where the sum runs over all examples of the training set.

The problem becomes a quadratic constrained optimization problem:

$$\text{Maximize with respect to } \alpha: \ (\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \gamma_i \gamma_j (x^i \cdot x^j)$$

$$\text{under the constraints}$$

$$\sum_{i=1}^{N} \alpha_i \gamma_i = 0$$

$$\alpha_i \geq 0 \ \forall i$$

This guarantees that all examples are correctly classified, and that the examples that lie closest to the separating hyperplane are as far as possible from the latter; in other words, the *margin* of the classifier, i.e., the distance between the separating surface and the examples is as large as possible (Figure 8).

It is shown that the only non-zero parameters α_i pertain to the examples of the training set that lie exactly on the margin (the *support vectors*), i.e., are located closest to the separating surface. Therefore, the number of nonzero parameters is usually much smaller than the number of examples, a straightforward consequence of the regularization mechanism present in the definition of the SVM.

If the examples are not linearly separable, the dot product $(\mathrm{x}_i\ \mathrm{x}_j)$ can be replaced by an appropriate *kernel function* $K\ (\mathrm{x}_i, \mathrm{x}_j)$, which is equivalent to defining a new *feature space* $\mathbf{z} = \phi(\mathbf{x})$ such that $\phi(\mathbf{x}_i)\ \phi(\mathbf{x}_j) = \boldsymbol{K}(\mathbf{x}_i, \mathbf{x}_j)$. If the training examples are linearly separable in the new feature space, the SVM machinery can be applied exactly as described above. The most popular kernel is the Gaussian kernel $K(\mathbf{x}, \mathbf{y}) = \exp \frac{\|\mathbf{x} - \mathbf{y}\|^2}{\sigma^2}$; the width σ is a hyperparameter whose value is found by cross-validation as described below.

Finally, if no satisfactory kernel can be found, the constraint that all training examples are correctly classified can be relaxed (*soft-margin SVM*). As a result, the last constraint of the dual form of the optimization problem becomes

$$0 < \alpha_i < C \ \forall i$$

where C is a hyperparameter, termed *regularization constant*, whose value is found by cross-validation. The larger the value of C, the more stringent the constraint of correct classification of all examples. The number of support vectors is equal to the number of examples that lie within the margin of the classifier; in the present study, about 25% of the training examples were found to be support vectors.

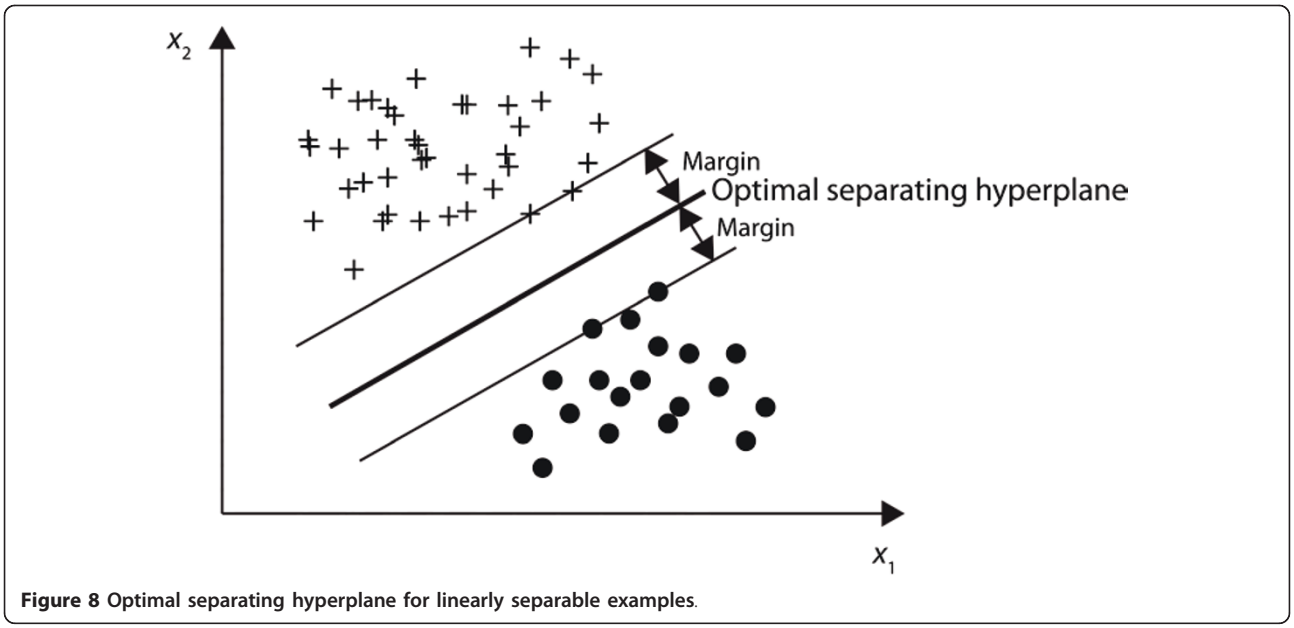

Figure 8 Optimal separating hyperplane for linearly separable examples.

Cross-validation consists of the following procedure: the available dataset is divided into two disjoint subsets: a *training/validation set* and a *test set*.

The training/validation set is in turn divided into D disjoint subsets or *folds*. A classifier is trained on D - 1 folds, and the resulting classifier is applied to the examples present in the remaining fold. The number of classification errors on these examples is stored in memory, and the procedure is iterated D times, so that each example in the training/validation set is present once and only once in a validation set. The *validation score* is the overall classification error rate, computed by counting the error on all examples *in the validation sets*, thereby providing an estimate of the performance of the classifier. The same procedure is repeated for various values of the hyperparameters, and the combination of the hyperparameters giving the classifier with the smallest validation score is retained. Finally, a classifier with the optimal hyperparameter combination is trained with all examples of the training/validation set; its performance is subsequently assessed on the test set, whose examples have never been used before, thereby providing a statistically valid estimate of the classifier performance.

Data preprocessing by principal component analysis

Principal components analysis is a useful preprocessing technique for finding a representation of the variables that is more compact than the representation used initially. Consider the following extreme case, illustrated on Figure 9: in *representation space*, i.e., in the space whose dimension is equal to the number of variables (three for graphical simplicity), each dot represents the values of the variables measured for an example of the training set. If the dots are aligned, it is clear that the problem, which seemed to be three-dimensional, can actually be described by a single variable: the abscissa of each point along the line, which is a linear combination of the primary variables. The PCA technique, based on the diagonalization of the covariance matrix of the variables, finds the parameters of that combination. More generally, if the variables have the structure of an elongated cloud, the data can be represented more compactly by a smaller number of variables that are linear combinations of the primary variable. The first principal axis found by the PCA procedure is the axis along which the variance of the primary variables is maximum, the second principal axis is the axis along which the remaining variance is maximum, etc. These axes are mutually orthogonal.

In this study, when PCA was used, classifiers with an increasing number of principal components were trained in succession; the error rate of each classifier was computed on a validation set, until the addition of a new principal component did not increase the validation score significantly. As shown in Tables 1 and 2, 4 to 10 principal components were found useful in our study.

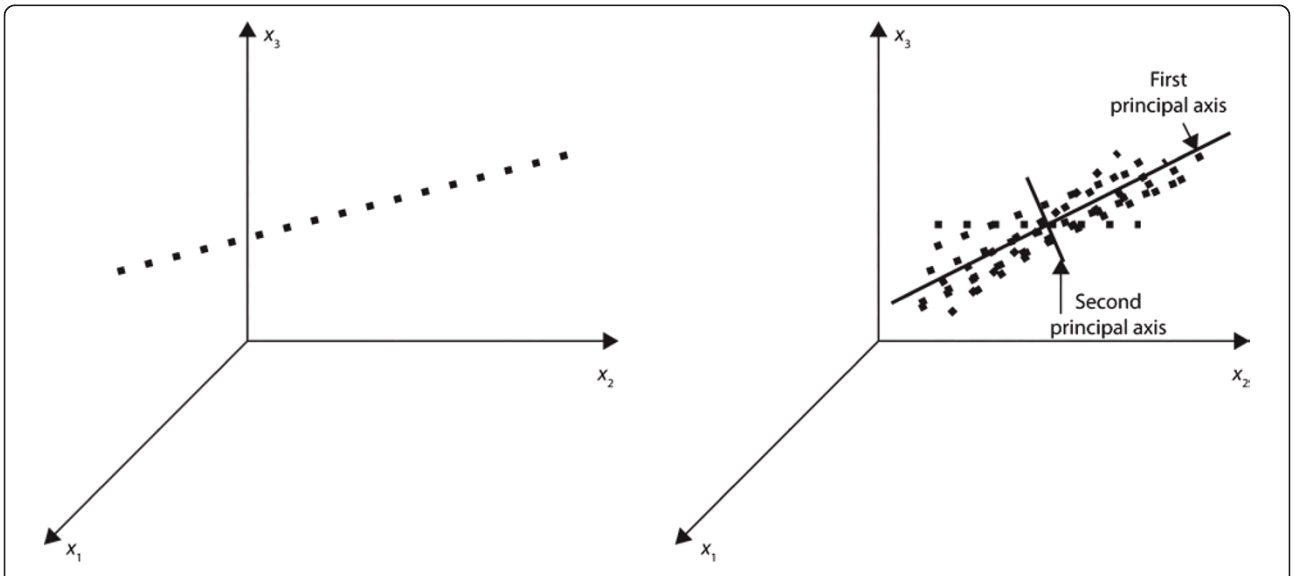

Figure 9 Graphical illustration of Principal Component Analysis in the case of a 3-dimensional representation space. Each dot shows the values of the variables pertaining to a given example. Left: all dots fall on a straight line, which means that each example can be described unambiguously by its abscissa along that line: the number of variables can be reduced can be reduced from 3 to 1 without information loss. Right: with negligible information loss, each example can be described by its coordinates with respect to the first two principal axes, so that the number of variables can be reduced from 3 to 2.

PCA should not be confused with variable selection procedures that assess the *relevance* of the variables, since PCA takes into account the variables only, and does *not* take into account the quantity to be predicted, i.e., the class of the item to be classified.

List of abbreviations

BTS: base transceiver station; CV: cross-validation; *K*-NN: *K*-nearest neighbor; OEM: original equipment manufacturer; PCA: principal components analysis; RSS: received signal strength; SVM: support vector machines.

Acknowledgements

The authors wish to acknowledge the reviewers and the editor-in-chief for numerous comments and suggestions for improving our article. They also acknowledge contributions by Rémi Dubois of Sigma Laboratory and the numerous research interns who contributed to the project over the past few years.

Author details

[1]Signal Processing and Machine Learning Laboratory, ESPCI - ParisTech, 10 rue Vauquelin, 75005 Paris, France [2]Université Pierre et Marie Curie, 4 place Jussieu, 75005 Paris, France

Competing interests

The authors are the inventors of patent FR2946825 (priority 2009-06-12) held by the Universite Pierre et Marie Curie.

Received: 30 November 2010 Accepted: 31 August 2011
Published: 31 August 2011

References

1. A Küpper, in *Location-Based Services: Fundamentals and Operation* (John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2005)
2. AM Ladd, KE Bekris, A Rudys, LE Kavraki, DS Wallach, On the feasibility of using wireless ethernet for indoor localization. IEEE Trans Robot Autom. **20**(3), 555–559 (2004). doi:10.1109/TRA.2004.824948
3. M Brunato, R Battiti, in *Statistical Learning Theory for Location Fingerprinting in Wireless LANs, Computer Networks and ISDN Systems April 2005*, vol. 47. (Elsevier Science Publishers, Amsterdam, 2005)(6), pp. 825–845
4. Q Yang, S Jialin Pan, V Wenchen Zheng, Estimating location using Wi-Fi. IEEE Intell Syst. **23**(1), 8–13 (2008)
5. S-H Fang, T-N Lin, P-C Lin, Location fingerprinting in a decorrelated space. IEEE Trans Knowledge Data Eng. **20**(5), 685–691 (2008)
6. S-H Fang, T-N Lin, Indoor location system based on discriminant-adaptive neural network in IEEE 802.11 environments. IEEE Trans Neural Netw. **19**(11), 1973–1978 (2008)
7. S-H Hong, B-K Kim, D-S Eom, Localization algorithm in wireless sensor networks with network mobility. IEEE Trans Consum Electron. **55**(4), 1921–1928 (2009)
8. S-P Kuo, Y-C Tseng, A scrambling method for fingerprint positioning based on temporal diversity and spatial dependency. IEEE Trans Knowledge Data Eng. **20**(5), 678–684 (2008)
9. H Lee, S Lee, Y Kim, H Chong, Grouping multi-duolateration localization using partial space information for indoor wireless sensor networks. IEEE Trans Consum Electron. **55**(4), 1950–1958 (2009)
10. D Zimmerman, J Baumann, M Layh, F Landstorfer, R Hoppe, G Wölfle, database correlation for positioning of mobile terminals in cellular networks using wave propagation models, in *Proceedings of IEEE 60th Vehicular Technology Conference, 26-29*, **7**, 4682–4686 (September 2004)
11. V Otsason, A Varshavsky, A LaMarca, E de Lara, Accurate GSM indoor localization, in *Proceedings of the 7th International Conference on Ubiquitous Computing, UbiComp 2005*, ed. by M Beigl (Springer-Verlag, Berlin, Heidelberg, 2005), pp. 141–158
12. Z Wu, C Li, JK-Y Ng, KRPH Leung, Location estimation via support vector regression. IEEE Trans Mob Comput. **6**(3), 311–321 (2007)
13. B Denby, Y Oussar, I Ahriz, in *Geolocalisation in Cellular Telephone Networks, Proceedings of the NATO 2007 Advanced Study Institute on Mining Massive DataSets for Security*, ed. by F Fogelman-Soulié, D Perrotta, J Piskorski, R Steinberger (IOS Press, Amsterdam, The Netherlands, 2008)
14. W ur Rehman, E de Lara, S Saroiu, CILoS, a CDMA indoor localization system, in *Proceedings of the 10th International Conference on Ubiquitous Computing, UbiComp 2008*, (Seoul, Korea, September 2008), pp. 21–24
15. B Denby, Y Oussar, I Ahriz, G Dreyfus, High-performance indoor localization with full-band GSM fingerprints, in *Proceedings IEEE International Conference on Communications, Workshop on Synergies in Communication and Localization (SyCoLo)*, (Dresden, Germany, June 2009)
16. I Ahriz, Y Oussar, B Denby, G Dreyfus, Carrier relevance study for indoor localization using GSM, in *Proceedings of the 7th Workshop on Positioning, Navigation and Communication 2010*, (Dresden, Germany, March 2010), pp. 11–12
17. I Ahriz, Y Oussar, B Denby, G Dreyfus, Full-band GSM fingerprints for indoor localization using a machine learning approach. International Journal of Navigation and Observation. **2010**, 7. Article ID 497829
18. N Cristianini, J Shawe-Taylor, in *Support Vector Machines and Other Kernel-Based Learning Methods*. (Cambridge University Press, Cambridge, 2000)
19. G Dreyfus, *Neural Networks Methodology and Applications*, (Springer, Springer-Verlag Berlin Heidelberg, 2005)
20. Tems Mobile System, http://www.ericsson.com/solutions/tems/
21. Telit GM862-GPS module, http://www.telit.com/en/products/gsm-gprs.php
22. Y-C Ho, RL Kashyap, An algorithm for linear inequalities and its applications. IEEE Trans Electron Comput. **14**(5), 683–688 (1965)
23. The Spider, http://www.kyb.tuebingen.mpg.de/bs/people/spider/
24. TM Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans Electron Comput. **14**, 326–334 (1965)
25. O Chapelle, B Schölkopf, A Zien, in *Semi-Supervised Learning* (MIT Press, Cambridge, MA, 2006)
26. X Zhu, Semi-Supervised Learning Literature Survey, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, Technical Report 1530. http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf (2006)
27. T Joachim, Transductive inference for text classification using support vector machines, in *International Conference on Machine Learning (ICML)*, pp. 200–209 (June 1999)
28. J Jia, L Cai, A TSVM-based minutiae matching approach for fingerprint verification, in *International Workshop on Biometric Recognition Systems (IWBRS)*, pp. 85–94 (October 2005)
29. SVM^{light}, http://svmlight.joachims.org/
30. J Wang, X Shen, W Pan, On transductive support vector machines. Contem Math, 443: 7–19 (2007)

doi:10.1186/1687-1499-2011-81
Cite this article as: Oussar *et al.*: Indoor localization based on cellular telephony RSSI fingerprints containing very large numbers of carriers. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:81.