Modeling the Received Signal Strength Intensity of Wi-Fi signal using Hidden Markov Models

Óscar Belmonte-Fernández[a,b,*]

[a]*Department of Computer Languages and Systems, Universitat Jaume I, Castelló de la Plana, 12071, Spain*
[b]*Institute of New Imaging Technologies, Universitat Jaume I, Castelló de la Plana, 12071, Spain*

Abstract

Wi-Fi fingerprinting is one of the methods that are widely used to provide Location Based Services (LBS). Gaussian, or a mixture of Gaussians, is the preferred model used by Wi-Fi fingerprinting for LBS. Nevertheless, Received Signal Strength Intensity (RSSI) Wi-Fi histograms are skewed, and a Gaussian model is not well suited for modeling data when their histogram is skewed. In addition, another important characteristic present in the RSSI Wi-Fi temporal series is autocorrelation, which cannot be modeled using a Gaussian model. In this paper, we explore the feasibility of using Hidden Markov Models (HMM) to model RSSI Wi-Fi signals. The mathematical derivation of formulas to calculate autocorrelation based on the HMM parameters is presented. Exhaustive experimentation, using data sampled in a real scenario, was performed to test the dependency of the autocorrelation coefficients on the number of hidden states, and the number of iterations used when creating the HMM. The results are compared with autocorrelation coefficients calculated using the real data. Kullback-Leibler (KL) divergence was used to compare the similarity of the real histograms and those provided by a mixture of Gaussians and by an HMM. HMM models reported more accurate results than a mixture of Gaussians model in both cases.

[*]Óscar Belmonte-Fernández
Email address: `oscar.belmonte@uji.es` (Óscar Belmonte-Fernández)

Keywords: Hidden Markov Models, RSSI Wi-Fi modeling, autocorrelation, mixture of Gaussians

1. Introduction

Indoor location research and indoor-based services have attracted a lot of attention in recent years, both in the research realm (Farid et al., 2013) and in the enterprise realm (Werner, 2014). Indoor location has been successfully used in different fields such as tourism and cultural applications (Alletto et al., 2015), location in malls (Pritt, 2013) and airports (Molina et al., 2018), and tele-health monitoring (Santoso & Redmond, 2015), to cite just a few examples. Various technologies have been used to provide indoor location services ranging from Bluetooth Low Energy (BLE) (Pu & You, 2018) to sound/ultrasound (Cobos et al., 2016). Nevertheless, indoor location based on Wi-Fi fingerprinting has attracted a great deal of research effort in the last few years (He & Chan, 2015; Yang & Shao, 2015). This is mainly due to the ubiquitous deployment of Wi-Fi technology, especially in urban areas, and the fact that consumer devices, such as mobile phones and smart-watches, are equipped with the hardware needed to use this technology.

The Gaussian probability density function (*pdf*), or a mixture of Gaussian *pdfs*, is the representation assumed in most works when modeling RSSI Wi-Fi histograms (Smailagic et al., 2000; Haeberlen et al., 2004; Bose & Foh, 2007; Fang et al., 2008; Qi et al., 2009; Pritt, 2013; Bisio et al., 2016; Youssef & Agrawala, 2008). This assumption is usually taken based on the similarity of the Wi-Fi RSSI histogram to a Gaussian distribution, but the similarity is not validated using any normality test in most cases.

Another issue when modeling RSSI Wi-Fi signals is the absent readings when signals coming from different Wireless Access Points (WAP) are measured at the same time. These absent values in the readings are due to low RSSI values when the distance between the WAP and the users is high, or when there is a lot of absorption by the walls and furniture present in the environment. Values below -95 dBm. are not measured by common mobile phones or smart-watches.

However, the main drawback when using a Gaussian *pdf*, or mixture of Gaussians, to model the RSSI Wi-Fi signal is that it cannot model the autocorrelation characteristic present in the temporal series of the Wi-Fi signal. An alternative model that maintains the autocorrelation present in the RSSI Wi-Fi signal is the Hidden Markov Model (HMM), which can be used to model temporal series when autocorrelation is present in the data (Zucchini et al., 2017). HMM has been successfully used in bioinformatics (Koski, 2001), speech recognition (Rabiner, 1989), and stock market prediction (Gupta & Dhingra, 2012).

The main hypothesis of this work is that an RSSI Wi-Fi signal can be modeled in a better way using an HMM, which is able to maintain the autocorrelation present in the signal better than a Gaussian, or mixture of Gaussians, model. Once the RSSI Wi-Fi signal has been modeled, the HMM can be used to make predictions on the new Wi-Fi readings, or to calculate the probability that a temporal sequence of RSSI data has been generated by the HMM model.

The contributions of this work are:

1. We present the mathematical formulation for calculating the autocorrelation coefficients based on the parameters of an HMM.
2. We model RSSI Wi-Fi signals by means of an HMM and compare its autocorrelation coefficient with the autocorrelation coefficients present in the signal.
3. We compare the histograms of the real data with the histograms provided by an HMM and a mixture of Gaussians when modeling a Wi-Fi signal.
4. We show that an HMM can deal with absent values when sampling Wi-Fi signals.

Theoretical derivations and extensive experimentation were performed in order to assess the above contributions.

The rest of the paper is organized as follows: Section 2 summarizes related work. Section 3 presents the theoretical background on Markov chains and HMMs. Section 4 shows how to calculate the autocorrelation coefficients and the histogram using the parameters of an HMM, and how an HMM can manage absent values in a sequence of RSSI Wi-Fi readings. Section 5 presents the experimentation setup, data acquisition and analysis, results of exhaustive experimentation to assess the preservation of the autocorrelation coefficients, and a comparison between the histograms of the real data and those generated by a mixture of Gaussians and by an HMM. Finally, Section 6 presents the conclusions and lines of future work.

2. Related work

Wi-Fi fingerprinting methods make use of the already deployed wireless communication infrastructures based on the IEEE 802.11 protocol. Wi-Fi fingerprinting generally consists of two stages: the off-line or training stage, and the on-line or location estimation stage. During the off-line stage, RSSI Wi-Fi signals coming from the surrounding WAPs are sampled at different reference locations within the zone of interest. Then, these data are used to create models that will be used in the on-line phase to estimate the user's location. In the on-line phase, the model will provide a location estimation based on the new RSSI Wi-Fi signal sampled by the user.

Some Wi-Fi fingerprinting methods model the RSSI Wi-Fi signal histogram using a *pdf*. The preferred *pdf* is the Gaussian function, defined by the mean and standard distribution of the data.

In (Smailagic et al., 2000), based on the histogram of a series of RSSI Wi-Fi samples, a Gaussian *pdf* is assumed in the off-line stage. No statistical test is performed to assess the normality of the data. In the on-line stage, using a path loss model and the Gaussian *pdf* assumption of the data distribution, the authors were able to estimate the user's location.

In (Haeberlen et al., 2004) a set of 255 RSSI Wi-Fi samples are used to fit a Gaussian *pdf*, in the off-line stage. A Bayesian approach is then used to estimate the user's location in the on-line stage. The accuracy of the method is compared with the results when the histogram of the real data are used to estimate the user's location, concluding that the Gaussian *pdf* fit works well. No normality test is performed to assess the normality distribution of the data.

The same approach is used by the authors in (Bose & Foh, 2007), who found through experimentation that the RSSI Wi-Fi signal histogram can be modeled by a Gaussian *pdf* when there is line of sight (LOS) between the WAP and the user, and also when there is no LOS (non-LOS). As in the previous work, no normality test is performed to assess the normality of the data distribution.

In (Fang et al., 2008), the authors model the RSSI Wi-Fi signal using a Gaussian fit in the off-line stage, and then use the logarithmic space to remove the convolution in the data due to multipath propagation of the signal. In the on-line stage, Maximum Likelihood Estimation (MLE) is used on the deconvoluted signal to estimate the user's location.

Although in (Qi et al., 2009) the authors model the RSSI Wi-Fi signal as a Gaussian *pdf* in the off-line stage, the approach used to estimate the user's location is different from that employed in the previously presented works. In this case, several samples are taken by the user and they are fitted to a Gaussian *pdf*. Then, using Machine Learning algorithms, the parameters of the user's fitted Gaussian *pdf* are used to estimate the user's location in the on-line stage. Again, no test is performed to assess the validity of assuming a Gaussian *pdf* for modeling the data.

In (Pritt, 2013) the authors fit RSSI Wi-Fi data to a Gaussian *pdf* in the off-line stage, and use MLE to estimate the user's location in the on-line stage. The novelty, in the context of this paper, is that low signals that vanish in some samples are modeled as a Gaussian *pdf* with mean -95 dBm. and standard deviation of 4 dBm., so they can be included in the MLE calculations.

In (Bisio et al., 2016) the authors use a similar approach to that implemented in (Qi et al., 2009). They model the RSSI Wi-Fi signal as a Gaussian *pdf*, and also the samples taken by the user in the on-line stage of the localization. In this case, the computation is simplified, while maintaining accuracy, in order to save energy consumption by the mobile device.

In (Li et al., 2018), instead of the Gaussian *pdf*, the Weibull *pdf* is used to model the RSSI Wi-Fi signal for user location purposes using a Bayesian approach. The rationale behind choosing the Weibull *pdf* is grounded on the fact that this *pdf* is able to model skewed histograms.

In (Youssef & Agrawala, 2008) the authors present the HORUS location system for location determination. They model the autocorrelation in the RSSI Wi-Fi data series using a Gaussian fit and a first-order autoregressive model. Taking into account the autocorrelation present in the temporal data series enables the authors to improve the accuracy of their location system.

Autocorrelation in the RSSI Wi-Fi temporal series appears in (Kaemarungsi & Krishnamurthy, 2004, 2012) as an important characteristic of the RSSI Wi-Fi signal to be modeled, but this is only used by the authors to assess the stability of the signal over time. In addition, the authors show that RSSI Wi-Fi histograms are, in general, left-skewed and that a Gaussian fit would not be valid in most cases.

These previous works show that although the *pdf* that is most used to model RSSI Wi-Fi signals is the Gaussian *pdf*, there are other alternatives for modeling it. They also demonstrate that the autocorrelation present in the RSSI Wi-Fi temporal series can be used to better model it.

3. Theoretical background

This section first presents the definition of a Markov chain and the existence of a stationary state for irreducible and homogeneous Markov chains. This property will be used in Section 4. Then the HMM is defined, together with a description of how its parameters can be estimated using data coming from a time sequence, which is the case of an RSSI Wi-Fi signal.

3.1. Markov chains

A stochastic process of random variables $\{q_1, q_2, q_3 ...\} = \{q_t : t \in T\}$, where $T = \{1, 2, 3, ...\}$ is a time index, with $q_t \in \{S_1, S_2, ..., S_N\} = \{S_j : j \in [1, N]\}$, called the state space, is a Markov chain if each state q_{t+1} depends only on the current state q_t, which is called the Markov property and can be expressed as follows:

$$
\begin{aligned}
&P(q_{t+1} = S_{j+1}|q_t = S_j, q_{t-1} = S_{j-1}, ..., q_1 = S_1) = \\
&P(q_{t+1} = S_{j+1}|q_t = S_j)
\end{aligned}
$$

A Markov chain is said to be homogeneous if the transition probabilities between states $a_{ij} = P(q_{t+1} = S_j|q_t = S_i)$ are independent of time t, in which case the probabilities can be represented by an $N \times N$ matrix:

$$
A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,N} \\ a_{2,1} & a_{2,2} & \dots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \dots & a_{N,N} \end{bmatrix}
$$

where each row follows the probability condition $\sum_{j=1}^{N} a_{ij} = 1, \forall i$. It can easily be proven that the transition probabilities after k steps is given by A^{k-1}. State S_j is said to communicate with state S_j (written $S_i \rightarrow S_j$) if the chain can at some time visit state S_j with a positive probability starting from state S_i. Furthermore, it is said that states S_i and S_j are interconnected (written $S_i \leftrightarrow S_j$) if $S_i \rightarrow S_j$ and $S_j \rightarrow S_i$. A Markov chain is defined as being irreducible if $S_i \leftrightarrow S_j$ for all $S_i, S_j \in \{S_1, S_2, \dots, S_N\}$.

Given an irreducible and homogeneous Markov chain with the following initial probability distribution:

$$
\begin{aligned}
\boldsymbol{\delta}(t = 1) &= (\delta_1(t = 1), \delta_2(t = 1), \dots, \delta_N(t = 1)) \\
&= (P(q_1 = S_1), P(q_1 = S_2), \dots, P(q_1 = S_N))
\end{aligned}
$$

with the probability condition $\boldsymbol{\delta}\mathbf{1'} = 1$, where $\mathbf{1'}$ is a column vector of ones, it is easy to calculate that after $t = k$ state transitions the probability distribution will be:

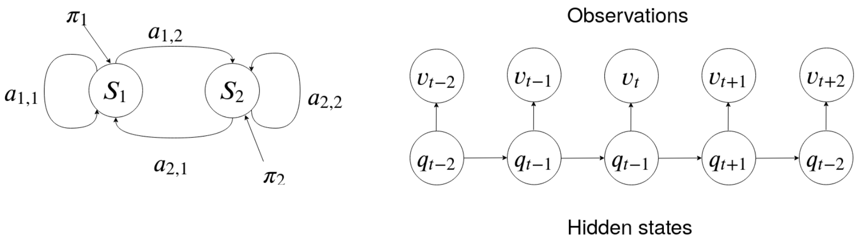


Figure 1: On the left, a graphical representation of an HMM with two hidden states. On the right, temporal evolution of an HMM.

$$\begin{aligned}\boldsymbol{\delta}(t=k) &= (\delta_1(t=k), \delta_2(t=1), \ldots, \delta_N(t=1)) \\ &= (P(q_k = S_1), P(q_k = S_2), \ldots, P(q_k = S_N)) \\ &= \boldsymbol{\delta}(1)\boldsymbol{A}^{k-1}\end{aligned}$$

It can be proven (Seneta, 2006) that when $t \to \infty$, $\boldsymbol{\delta} = \boldsymbol{\delta}(t \to \infty) = \boldsymbol{\delta}(1)\boldsymbol{A}^{t\to\infty}$ converges to a fixed unique vector $\boldsymbol{\delta}$, which is called the stationary distribution of the probability matrix $\boldsymbol{A}$, and the Markov chain is also said to be stationary, that is:

$$\boldsymbol{\delta A} = \boldsymbol{\delta} \quad \text{with} \quad \boldsymbol{\delta 1}' = 1 \tag{1}$$

We will use the stationary distribution in Section 4 when modeling the stationary autocorrelation and histogram of an HMM that models the RSSI Wi-Fi signal.

3.2. Hidden Markov models

Following the presentation of HMM given in (Rabiner, 1989), an HMM is characterized by:

1. The number of hidden states N. An individual state is denoted as $S \in \{S_1, S_2, ..., S_N\}$, and the state at time t is q_t.
2. The number of different observation symbols M. An individual symbol is denoted as $V \in \{V_1, V_2, ...V_M\}$.

3. The probability distributions matrix for transitions between two states $\boldsymbol{A} = \{a_{ij}\}$, where $a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N$.
4. The probability distribution vector for observing a symbol in state $j, \boldsymbol{B} = \{b_j(k)\}$, where $b_j(k) = P(v_k \ at \ t | q_t = S_j), \quad 1 \leq j \leq N, \ 1 \leq k \leq M$.
5. The probability distributions vector for initial states $\boldsymbol{\pi} = \{\pi_i\}$ where $\pi_i = P(q_1 = S_i), \quad 1 \leq i \leq N$.

Note that characteristics 1 and 3 are the same as for the definition of a Markov chain. Note too that due to characteristic 4, the probability of observation $b_j(k)$ only depends on states $q_t = S_j$ and it is independent of any other previous state $q_{t-1}, q_{t-2}, ..., q_1$. An HMM can be compactly represented as $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$. Thus, an HMM is composed of two processes, a Markov chain which determines the state at time t ($q_t = S_i$), and a stochastic process at each state $j \in \{1, 2, \ldots, N\}$, which generates the observations with probability distribution $j, \boldsymbol{B} = \{b_j(k)\}$. In this work only homogeneous and irreducible HMMs are considered.

Figure 1, on the left, represents an HMM with two states. Each state is represented by a node which contains the probability distribution $b_j(k)$ of observing each symbol $V \in \{V_1, V_2, ...V_M\}$. Each directed edge represents a transition between two states with probability distribution a_{ij}. The arrows represent the probability distribution for the initial states π_i. The emissions v_t for each state q_t and the underlying Markov chain are shown on the right of Figure 1.

An interesting question is (Problem 3 in (Rabiner, 1989)) how to find the HMM $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ which maximizes $P(O|\lambda)$ for a given number of hidden states N, a given number of observation symbols M, and a sequence of observations $O = O_1, O_2, ..., O_T; \ O_i \in V$. One solution to this problem is the Baum-Welch algorithm, which is a kind of Expectation-Maximization (EM) method that uses the forward and backward probabilities to estimate A and B (see references: (Baum et al., 1970; Rabiner, 1989; Jurafsky, 2000)). The Baum-Welch algorithm has been used to create all the HMMs implemented in this work.

4. Modeling RSSI Wi-Fi signals using HMM

This section first presents how to calculate the autocorrelation coefficients in a sequence of observations $O = O_1, O_2, ..., O_T;\ O_i \in V$ from the parameters of the HMM that models such a sequence. Then, the method used to calculate the probability histogram from the parameters of the HMM is described. Finally, the issue of absent values in the sequences of data is presented along with how an HMM can manage these absent values.

4.1. Autocorrelation for HMM

The autocorrelation ρ_l of a stochastic process $O = O_1, O_2, \ldots, O_T$ with observations $O_i \in V$ of length T for a lag l is the correlation between values of the process at different lag l times and it is calculated using the following formula (Murphy, 2012):

$$\rho_l = \frac{\frac{1}{T-l}\sum\limits_{t=1}^{T-l}(O_t - \bar{O})(O_{t+l} - \bar{O})}{\frac{1}{T-1}\sum\limits_{t=1}^{T}(O_t - \bar{O})^2} \tag{2}$$

The denominator of the above formula is the variance of the sequence of observations O. This formula can be rewritten as:

$$\rho_l = \frac{\sum\limits_{(V_i,V_j)\in V}(V_i - \bar{O})(V_j - \bar{O})\left[\frac{1}{T-l}\sum\limits_{t=1}^{T-l}P(V_i(t+l), V_j(t))\right]}{\sum\limits_{V_i}^{V_i\in V}(V_i - \bar{O})^2\left[\frac{1}{T-1}\sum\limits_{t=1}^{T}P(V_i(t))\right]} \tag{3}$$

where $P(V_i(t+l), V_j(t))$ is the joint probability distribution of observation $O_{t+l} = V_i$ at time $t + l$ and observation $O_t = V_j$ at time t regardless of the states at time $t + l$ and t, respectively. The above formula is computationally interesting because the sum of observations O_i and O_j is independent of the sum of the time t and the lag l, so no recalculation of the first sum is needed if the lag l is changed. If we define $\boldsymbol{V}$ as the matrix:

$$\boldsymbol{V} = \begin{bmatrix} (V_1-\bar{O})(V_1-\bar{O}) & (V_1-\bar{O})(V_2-\bar{O}) & \dots & (V_1-\bar{O})(V_M-\bar{O}) \\ (V_2-\bar{O})(V_1-\bar{O}) & (V_2-\bar{O})(V_2-\bar{O}) & \dots & (V_2-\bar{O})(V_M-\bar{O}) \\ \vdots & \vdots & \ddots & \vdots \\ (V_M-\bar{O})(V_1-\bar{O}) & (V_M-\bar{O})(V_2-\bar{O}) & \dots & (V_M-\bar{O})(V_M-\bar{O}) \end{bmatrix}$$

and $\boldsymbol{P}_l$ as the matrix:

$$\boldsymbol{P}_l = \tfrac{1}{T-l} \begin{bmatrix} \sum\limits_{t=1}^{T-l} P(V_1(t+l), V_1(t)) & \sum\limits_{t=1}^{T-l} P(V_1(t+l), V_2(t)) & \dots & \sum\limits_{t=1}^{T-l} P(V_1(t+l), V_M(t)) \\ \sum\limits_{t=1}^{T-l} P(V_2(t+l), V_1(t)) & \sum\limits_{t=1}^{T-l} P(V_2(t+l), V_2(t)) & \dots & \sum\limits_{t=1}^{T-l} P(V_2(t+l), V_M(t)) \\ \vdots & \vdots & \ddots & \vdots \\ \sum\limits_{t=1}^{T-l} P(V_M(t+l), V_1(t)) & \sum\limits_{t=1}^{T-l} P(V_M(t+l), V_2(t)) & \dots & \sum\limits_{t=1}^{T-l} P(V_M(t+l), V_M(t)) \end{bmatrix}$$

the autocorrelation in Equation 3 can be calculated by the formula:

$$\rho_l = sum(\boldsymbol{V} * \boldsymbol{P}_l^T) \tag{4}$$

where the $*$ operation is the element-by-element multiplication of the two matrices, and *sum* is the sum of all the elements in the resulting matrix.

The joint probability distribution $P(V_i(t+l), V_j(t))$ in the numerator of Eq. 3 can be calculated by applying the chain rule and summing up all the intermediate states, regardless of the intermediate emissions as follows:

$$\begin{aligned} P(V_i(t+l), V_j(t)) = &\sum_{S_k(t+l), S_k(t+l-1), \dots, S_k(t) \in S} P(V_i(t+l)|S_k(t+l))P(S_k(t+l)|S_k(t+l-1)) \\ &P(S_k(t+l-1)|S_k(t+l-2)) \dots P(S_k(t+1)|S_k(t))P(V_j(t)|S_k(t))P(S_k(t)) = \\ &\sum_{S_k(t+l), S_k(t) \in S} P(V_i(t+l)|S_k(t+l))\{\boldsymbol{A}^l\}_{i,j} P(V_j(t)|S_k(t))P(S_k(t)) \end{aligned}$$

where $P(V_i(t+l)|S_k(t+l))$ is the probability of emitting the observation V_i in state S_k and at time $t+l$. If we assume that the emission probability is independent of time, we get $P(V_i(t+l)|S_k(t+l) = P(V_i(t)|S_k(t) = P(V_i|S_k)$, which can be expressed as a matrix $\boldsymbol{O}$, where each element $\boldsymbol{O}_{ki} = P(V_i|S_k)$. Analogously, $P(S_1(t)), P(S_2(t)), \dots p(S_N(t))$ are the probability of being in state

$S_1, S_2, \ldots, S_N$ after time t, which can be represented as a vector as $\boldsymbol{S}(t) = \boldsymbol{\pi}\boldsymbol{A}^t$. Equation 4.1 can be rewritten using matrices as:

$$\begin{gathered} P(V_i(t+l), V_j(t)) = \boldsymbol{V}_{ij} = \{[\boldsymbol{O}\bar{*}\boldsymbol{S}(t)]^T\,\boldsymbol{A}^l\boldsymbol{O}\}_{ij} \\ \boldsymbol{V} = [\boldsymbol{O}\bar{*}\boldsymbol{S}(t)]^T\,\boldsymbol{A}^l\boldsymbol{O} \end{gathered}$$

where the operation $\boldsymbol{O}\bar{*}\boldsymbol{S}(t)$ multiplies each column vector in $\boldsymbol{O}$ element by element by the elements in vector $\boldsymbol{S}(t)$:

$$\boldsymbol{O}\bar{*}\boldsymbol{S}(t) = \begin{bmatrix} O_{1,1}S_1(t) & O_{2,1}S_1(t) & \ldots & O_{M,1}S_1(t) \\ O_{1,2}S_2(t) & O_{2,2}S_2(t) & \ldots & O_{M,2}S_2(t) \\ \vdots & \vdots & \ddots & \vdots \\ O_{1,N}S_N(t) & O_{2,N}S_N(t) & \ldots & O_{M,N}S_N(t) \end{bmatrix}$$

Note that for a long sequence of observations with $t \to \infty$ and using Equation 1, one can write:

$$\lim_{t\to\infty} \boldsymbol{S}(t) = \lim_{t\to\infty} \boldsymbol{\pi}\boldsymbol{A}^t = \boldsymbol{\delta} \tag{5}$$

and so:

$$\lim_{t\to\infty} P(V_i(t+l), V_j(t)) = \lim_{t\to\infty} \{[\boldsymbol{O}\bar{*}\boldsymbol{S}(t)]^T\,\boldsymbol{A}^l\boldsymbol{O}\}_{ij} = \{[\boldsymbol{O}\bar{*}\boldsymbol{\delta}]^T\,\boldsymbol{A}^l\boldsymbol{O}\}_{ij} = V_{ij} \tag{6}$$

This expression is independent of time t, and only depends on lag l. In addition, when $t \to \infty$ the limit for any element in matrix $\boldsymbol{P}_l$ is:

$$\lim_{T\to\infty} P(V_i(T+l), V_j(T)) = \lim_{T\to\infty} \frac{1}{T-l} \sum_{t=1}^{T-l} P(V_i(t+l), V_j(t)) = V_{ij} \tag{7}$$

The same rationale can be applied to the denominator of the autocorrelation in Equation 3. Finally, the autocorrelation for an infinite sequence of observations generated by an HMM can be written as:

$$\rho_l = \frac{\sum\limits_{(V_i,V_j)\in V} (V_i - \bar{O})(V_j - \bar{O})V_{ij}}{\sum\limits_{V_i}^{V_i\in V} (V_i - \bar{O})^2 V_{ii}} \tag{8}$$

This equation provides the autocorrelation coefficient for lag l based only on the parameters that define an HMM.

4.2. Probability histogram generated by a HMM

The probability histogram of a set of N observations for each element in the vocabulary, taking into account characteristic 4 of HMMs (see Section 3.2), is given by:

$$p_T(O_i) = \frac{1}{T}\sum_{t=1}^{T}\sum_{S_j(t)\in\{S_1,\ldots,S_N\}} p(O_t = O_i|S_j(t)) \tag{9}$$

For all elements in the vocabulary $V \in \{V_1, \ldots, V_M\}$, the following matrix expression can be used:

$$p_T(\boldsymbol{O}) = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\pi}\boldsymbol{A}^t\boldsymbol{O} \tag{10}$$

where $\boldsymbol{S}(t) = \boldsymbol{\pi}\boldsymbol{A}^t$. One could consider the probability histogram when $t \to \infty$ using Equation 5:

$$\lim_{t\to\infty} p_t(\boldsymbol{O}) = \boldsymbol{\delta}\boldsymbol{O} \tag{11}$$

Equation 11 has the following meaning: the probability distribution of observations in the stationary state is the probability distribution of observations in each state $\boldsymbol{O}$ weighted by the distribution function of the stationary state $\boldsymbol{\delta}$.

4.3. Absent values in the temporal series

When acquiring Wi-Fi measures in real cases, sometimes no measure might be obtained for a certain WAP. The main reasons explaining this behavior are: a) the strength of the Wi-Fi signal is too low to measure due to the long distance between the WAP and the point where the measurement was taken; b) the absorption suffered by the signal when traversing walls and furniture present in the environment causes the signal to disappear; c) the software or electronic circuits have been unable to perform the measurement at this time. Commonly, in the domain of Wi-Fi fingerprinting, absent values are substituted by an ad-hoc value, $-100dBm$ (see for example the histogram for **mac_6** in Figure 4). In some applications that use Wi-Fi signals to perform their tasks, such as indoor location (He & Chan, 2015; Torres-Sospedra et al., 2015) where machine learning algorithms based on some distance metrics are used, this ad-hoc value could have some impact on the final result.

When using an HMM, an absent value is just another symbol in the vocabulary of possible symbols present in the RSSI Wi-Fi signal; in fact, the vocabulary for an HMM can be completely replaced by a new one and the HMM will model the same behavior present in the observations.

Hence, it can be said that an HMM can work naturally with absent values present in the RSSI Wi-Fi signal.

5. Experimental Results

This section first presents the experimental setup, data acquisition, and data analysis. It then presents exhaustive experimentation conducted in order to study the dependency of the calculated autocorrelation coefficients with the number of iterations and the number of hidden states when creating the HMM. Finally, the histogram generated using a mixture of Gaussian *pdfs* and an HMM are compared with the histogram of the original data using the Kullback-Leibler (KL) divergence.

5.1. Data acquisition

The data for this work were acquired at the Department of Computer Languages and Systems, which belongs to Universitat Jaume I in Castelló de la Plana, Spain. Altogether 500 consecutive Wi-Fi samples were acquired. Not all

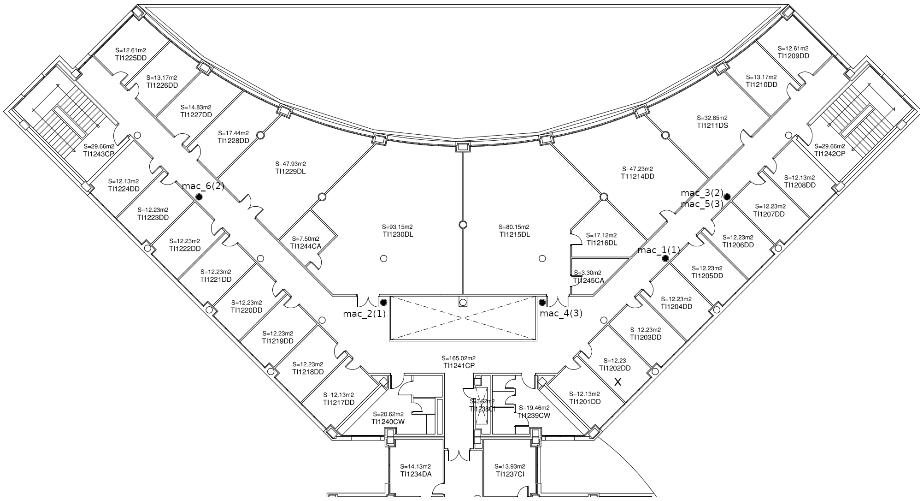


Figure 2: Map of the build where Wi-Fi samples were collected. The building is made of three plants and the basement. The map shows the location of the WAPs (black circles with plant number in parenthesis), and the laptop used to collect the data (black cross inside office with code TI1202DD).The three plants have similar offices distribution.

sampling attempts provided a value: in the case of mac_6, 403 out of a total of 500 sampling attempts were successful. These absent values were replaced by the value $-100dBm.$, which is the replacement value typically used in the domain of Wi-Fi fingerprinting. Only the signals coming from WAPs belonging to the *eduroam* Wi-Fi network were used. Signals coming from other Wi-Fi networks were filtered out. The *eduroam* network is an academic Wi-Fi network present on campuses and other research premises. ***Eduroam*** **WAPs points were used to avoid any bias when deploying ad-hoc networks for experimental purposes. Figure 2 presents a map showing the location of the WAPs and the laptop used to collect the data.** Table 1 shows the Mac addresses of the six WAPs used in this work, and statistics of the datasets thus acquired. The hardware used to acquire the RSSI Wi-Fi signals was a Lenovo Ideapad 330 laptop equipped with an Intel Wireless-AC 9560 chip. The software used was an ad-hoc Java application running with administration privileges.

Table 1: Characteristics of the datasets. The column with the heading #Samples shows the number of samples acquired for each Mac address. Mean and standard deviation are in dBm. Absent samples were not replaced by -100 dBm before calculating the mean and standard deviation

ID	Mac address	#Samples	Mean	Sd
mac_1	00:1A:6D:9B:9A:21	500	-69.92 ± 0.18	4.10 ± 0.13
mac_2	00:1A:A1:5C:F9:61	500	-70.92 ± 0.19	4.30 ± 0.14
mac_3	00:13:C3:44:D8:E1	500	-63.02 ± 0.19	4.24 ± 0.13
mac_4	00:13:1C:DD:FC:41	500	-60.58 ± 0.17	3.78 ± 0.12
mac_5	00:17:59:FB:19:51	500	-77.93 ± 0.13	2.87 ± 0.09
mac_6	00:13:C3:44:D6:71	403	-80.69 ± 0.13	2.56 ± 0.09

5.2. Data analysis

The column on the left of Figures 3 and 4 shows the histograms of 500 samples for each of the RSSI Wi-Fi signals analyzed. The superimposed curve is the Gaussian *pdf* fit of the data. For the case of mac_6, the $RSSI = -100$ values were excluded before the fit. The column on the right in the same figures shows the first ten autocorrelation values of the corresponding signals.

Each dataset was fitted to a Gaussian *pdf* using the R package *MASS*; the fitted parameters are shown in Table 1. To assess the goodness of each fit, the Pearson (Pearson, 1900), Anderson-Darling (Anderson & Darling, 1954), Shapiro-Wilk (Shapiro & Wilk, 1965), and Jarque-Bera (Jarque & Bera, 1987) normality tests were used. The Jarque-Bera normality test is robust regarding the number of samples used in the test, but the other tests used are less robust regarding the number of samples. To take this dependency into account, normality tests were performed on groups of 25, 50, 75 and 100 samples each. In addition, the normality tests were also performed on data sampled from the fitted Gaussian *pdf* using the same group sizes as for the real data, which allows the results for real and sampled data to be compared. For the Gaussian *pdf* fitted, real numbers where rounded to the nearest integer before applying the normality tests. Results for the p-value are shown in Tables 2 - 4, the header of each column is the number of consecutive samples used in the test; i.e., for

Table 2: p-values for the results of the Pearson normality tests. The column headers stand for the number of samples taken for each test. Each cell is for the mean p-value and its standard deviation for the sample sizes in each column.

	25	50	75	100
mac_1	0.1±0.2	0.04±0.1	0.02±0.03	2e-08±5e-08
normal	0.2±0.2	0.3±0.3	0.3±0.3	0.08±0.2
mac_2	0.05±0.09	0.008±0.02	1e-04±4e-04	9e-09±1e-08
normal	0.4±0.3	0.3±0.3	0.2±0.3	0.3±0.5
mac_3	0.02±0.04	0.03±0.08	0.02±0.05	0.002±0.005
normal	0.3±0.2	0.3±0.3	0.2±0.3	0.3±0.3
mac_4	0.05±0.1	0.01±0.03	0.004±0.01	1e-05±2e-05
normal	0.3±0.3	0.2±0.2	0.2±0.2	0.09±0.1
mac_5	0.02±0.03	9e-06±2e-05	2e-05±4e-05	5e-07±1e-06
normal	0.4±0.3	0.06±0.1	0.002±0.005	9e-07±2e-06
mac_6	9e-06±3e-05	4e-11±1e-10	8e-17±2e-16	2e-19±3e-19
normal	0.2±0.2	0.02±0.07	5e-06±8e-06	7e-13±9e-13

the column with header 25, 20 groups of 25 samples each were used, the first group with samples for indices from 1 to 25 in the temporal series, the second group with indices from 26 to 50, and so on. The value in each cell in Table 1 is the mean and standard deviation of the p-value. Taking the standard rejection p-value = 0.05 for mac_1, the p-values for all tests are compatible with the no normality assumption in all cases. For mac_2 and mac_3, the p-values for all tests are compatible with the no normality assumption, except for the Jarque-Bera test with a group size of 25 samples (p-value = 0.3 ± 0.2 in both cases). For mac_4, the p-values for all tests are compatible with the no normality assumption except for the Jarque-Bera test, which, on the contrary, is compatible with the normality assumption for all grouping sizes. For mac_5 and mac_6, the p-values for all tests are compatible with the no normality assumption.

Therefore, on the basis of the results provided by the normality tests performed, the assumption of normality on the RSSI Wi-Fi signals analyzed is very weak.

Table 3: p-values for the results of the Anderson-Darling normality tests. The column headers stand for the number of samples taken for each test. Each cell is for the mean p-value and its standard deviation for the groups of sample sizes in each column.

	25	50	75	100
mac_1	0.06±0.1	0.01±0.02	0.008±0.02	8e-06±1e-05
normal	0.5±0.3	0.3±0.1	0.1±0.1	0.2±0.2
mac_2	0.01±0.03	0.003±0.01	4e-04±9e-04	5e-06±1e-05
normal	0.3±0.2	0.2±0.2	0.2±0.2	0.2±0.04
mac_3	0.04±0.1	0.003±0.005	0.04±0.1	0.01±0.03
normal	0.4±0.3	0.3±0.3	0.2±0.1	0.2±0.1
mac_4	0.03±0.07	0.002±0.004	0.01±0.03	5e-04±0.001
normal	0.4±0.2	0.3±0.3	0.2±0.1	0.1±0.02
mac_5	0.03±0.05	0.006±0.01	8e-04±0.002	3e-04±2e-04
normal	0.3±0.2	0.2±0.1	0.09±0.05	0.02±0.02
mac_6	7e-05±3e-04	6e-06±1e-05	5e-05±1e-04	2e-05±3e-05
normal	0.2±0.2	0.1±0.1	0.05±0.04	0.02±0.01

5.3. Assessment in preserving the autocorrelation by HMM Wi-Fi modeling

Figures 3 - 4 show, in the column on the right, the autocorrelation coefficients for each RSSI Wi-Fi signal used in this work. For mac_2, mac_5 and mac_6, the first ten autocorrelation coefficients are greater than 0.2. For mac_3, the autocorrelation coefficients alternate between values greater than 0.4 and values close to 0.1. For mac_4, the autocorrelation coefficients alternate between positive and negative values. Finally, for mac_1, the autocorrelation coefficients are low but greater than 0.1 in all cases except for the autocorrelation coefficient in index 9, which is lower than 0.1. From all these figures a clear autocorrelation can be observed in the temporal series of the RSSI Wi-Fi signals.

In contrast, the samples generated by a normal distribution, or a mixture of normal distributions, are independent, and hence no autocorrelation is present in a sequence of data generated by a normal distribution. Figure 5 shows the histogram for 500 samples generated by a mixture of normal distributions (details on fitting the data to a mixture of normal distributions are presented in

Table 4: p-values for the results of the Shapiro-Wilk normality tests. The column headers stand for the number of samples taken for each test. Each cell is for the mean p-value and its standard deviation for the groups of sample sizes in each column.

	25	50	75	100
mac_1	0.08±0.2	0.02±0.05	0.02±0.04	1e-04±1e-04
normal	0.4±0.2	0.4±0.2	0.2±0.2	0.3±0.2
mac_2	0.02±0.05	0.007±0.02	6e-04±0.001	6e-05±1e-04
normal	0.4±0.3	0.3±0.3	0.4±0.4	0.4±0.4
mac_3	0.05±0.1	0.007±0.01	0.07±0.2	0.04±0.08
normal	0.4±0.3	0.4±0.2	0.4±0.3	0.1±0.1
mac_4	0.03±0.05	0.003±0.005	0.01±0.03	0.001±0.002
normal	0.5±0.3	0.4±0.2	0.3±0.2	0.3±0.2
mac_5	0.05±0.08	0.02±0.04	0.003±0.004	0.002±0.001
normal	0.4±0.2	0.2±0.2	0.1±0.1	0.1±0.2
mac_6	1e-04±2e-04	7e-05±1e-04	3e-04±5e-04	1e-04±2e-04
normal	0.3±0.2	0.1±0.1	0.1±0.1	0.04±0.04

Section 5.4) on the left, and its autocorrelation coefficients on the right. It can be observed that all autocorrelation coefficients are close to zero. The same result was obtained for the other RSSI Wi-Fi datasets used in this work.

In the following sections, we first present the dependency of the autocorrelation coefficients on the number of iterations used to create an HMM. We then present the dependency of the autocorrelation coefficients on the number of hidden states used to create an HMM.

5.3.1. Autocorrelation depending on the number of iterations

This section presents the results on the evolution of the autocorrelation coefficients depending on the number of iterations used to create an HMM. Figure 6 shows the results for each of the six WAPs used in the experiments. The number of hidden states used to create each HMM was 12 in all cases. The first ten coefficients were studied, labeled lag_1 to lag_10. Equation 4 was used to calculate the autocorrelation coefficients for an HMM, and a temporal

Table 5: p-values for the results of the Jarque-Bera normality tests. The column headers stand for the number of samples taken for each test. Each cell is for the mean p-value and its standard deviation for the groups of sample sizes in each column.

	25	50	75	100
mac_1	0.2±0.2	0.1±0.1	0.1±0.2	0.03±0.04
normal	0.7±0.2	0.6±0.3	0.7±0.2	0.6±0.3
mac_2	0.3±0.2	0.1±0.2	0.05±0.06	0.02±0.02
normal	0.6±0.3	0.6±0.3	0.4±0.3	0.4±0.4
mac_3	0.3±0.2	0.1±0.2	0.1±0.2	0.1±0.2
normal	0.6±0.2	0.6±0.3	0.6±0.3	0.5±0.5
mac_4	0.5±0.3	0.4±0.3	0.3±0.2	0.2±0.1
normal	0.6±0.2	0.6±0.3	0.6±0.3	0.6±0.2
mac_5	0.3±0.3	0.2±0.4	0.1±0.1	0.2±0.3
normal	0.6±0.2	0.5±0.3	0.5±0.2	0.8±0.2
mac_6	0.2±0.2	0.3±0.4	0.4±0.4	0.2±0.2
normal	0.6±0.3	0.5±0.4	0.7±0.2	0.5±0.4

series length of 500 samples. Each curve in a plot shows the evolution of one of the ten coefficients studied. The last column in each plot shows the value for the autocorrelation coefficients calculated using the real RSSI Wi-Fi temporal series.

In a general way, it can be said that the autocorrelation coefficients approach the real values when more iterations are used to create the HMM. For mac_4, the autocorrelation coefficients are close to 0 when the number of iterations used to create an HMM is lower than 50, but when the number of iterations is equal to or greater than 50, the coefficients quickly approach the real ones. For mac_1, the trend is much smoother when compared to mac_4. For mac_2, mac_5 and mac_6, there is a jump when the number of iterations increases from 20 to 30, followed by a plateau afterwards. The autocorrelation coefficients for mac_3 do not follow the pattern of any other WAP: some coefficients (lag_2 to lag_4) increase monotonically while others (lag_1, and lag_5 to lag_10) decrease monotonically.

5.3.2. Autocorrelation depending on the number of hidden states

This section presents the results on the evolution of the autocorrelation coefficients depending on the number of hidden states used to create an HMM. Six different HMMs were created with 6, 12, 25, 50, 75 and 100 hidden states, using the data acquired for each of the six WAPs, as described in Section 5.1.

Figures 7 to 12 show the results for each of the six WAPs used in the experiments. Each figure shows six different plots, one for each value of the number of hidden states in the set: 6, 12, 25, 50, 75 or 100. Each plot shows the evolution of the first ten autocorrelation coefficients, one curve for each coefficient, as the number of iterations ranges between 10 and 100 in steps of 10 iterations. The last column for each plot shows the autocorrelation coefficients from the real temporal series of RSSI Wi-Fi signals calculated using the formula in Equation 2. The autocorrelation coefficients for the HMM and 500 samples were calculated using Equation 4.

For mac_1 and 6 hidden states (Figure 7, the evolution of the autocorrelation coefficients provided by the HMM is far from the real autocorrelation coefficients regardless of the number of iterations used to create the HMM. In contrast, for 12 hidden states and 100 iterations, the coefficients are much closer to the real values than in the case of 6 hidden states. The same results can be observed in the evolution of the autocorrelation coefficients for plots when 25, 50, 75 and 100 hidden states were used to create the corresponding HMM.

For mac_2, mac_5 and mac_6 (Figures: 8, 11 and 12), even when the number of hidden states is 6, if the number of iterations used to create the model was greater than 50, the autocorrelation coefficients calculated for the corresponding HMM are close to the real autocorrelation coefficients.

The evolution patterns of the autocorrelation coefficients for mac_3 and mac_4 are not so clear as in the previous cases. But for mac_3 and mac_4 (Figures: 9 and 10), it can be seen that the number of iterations needed to create the corresponding HMM in order to provide autocorrelation coefficients

Table 6: Euclidean distance between the real autocorrelation coefficients and the autocorrelation coefficients for an HMM calculated with Equation 4 for each WAP studied. The Ratio column stands for the ratio between the module of the vector of real autocorrelation coefficients and the module of the vector of HMM autocorrelation coefficients. The Cosine column stands for the cosine between the vector of the real autocorrelation coefficients and the vector of the HMM autocorrelation coefficients.

WAP	States	Iterations	Distance	Ratio	Cosine
mac_1	50	80	0.04389	1.16251	0.91891
mac_2	75	60	0.02000	1.01248	0.99383
mac_3	75	90	0.05308	1.21710	0.98976
mac_4	6	100	0.08866	1.44270	0.97576
mac_5	75	60	0.00709	0.97476	0.99779
mac_6	50	70	0.00804	0.99312	0.99899

other than zero increases as the number of hidden states also increases.

Table 6 shows, for each WAP analyzed, the combination of the number of states and the number of iterations used to create an HMM with the shortest Euclidean distance between the real autocorrelation coefficients and the autocorrelation coefficients for the corresponding HMM. While the Euclidean distance is useful to compare the accuracy of the autocorrelation coefficients for the same HMM using different combinations of the number of states and the number of iterations, this distance is not so useful when comparing the autocorrelation coefficients provided by an HMM for RSSI Wi-Fi signals generated by different WAPs. In this last case, it can be useful to compare the ratio between the module of the 10-dimensional vector for the first ten real autocorrelation coefficients, and the autocorrelation coefficients provided by an HMM and also the cosine between the two vectors. The closer the ratio and the cosine are to 1, the greater the accuracy will be between real and HMM-provided autocorrelation coefficients. Table 6 shows, in addition to the Euclidean distance, the ratio between modules and the cosine between autocorrelation vectors for all RSSI Wi-Fi signals used.

For all WAP analyzed, the shortest Euclidean distance between the real autocorrelation coefficients and the autocorrelation coefficients provided by an HMM is 0.00709 for mac_5 with a combination of 50 hidden states and 80 iterations (see Table 6). But if the ratio between the vector modules and the cosine between vectors are used, the "closest" autocorrelation coefficients are for mac_6 with 50 hidden states and 70 iterations (ratio = 0.99312, cosine = 0.99899). Leaving to one side mac_4, which provides the highest Euclidean distance, the results in Table 6 show that the shortest distances are provided by an HMM having 50 or 75 hidden states, and when the number of iterations used to create them were between 60 and 90.

From the results shown in Table 6 for the RSSI Wi-Fi signal studied, it can be said that a good choice to create an HMM from an RSSI Wi-Fi temporal series is to choose between 50 and 75 hidden states and a number of iterations equal to or greater than 60, to preserve the autocorrelation present in the original signal.

5.4. Generated histograms

This section presents the accuracy between the histogram for the real data, and the histograms provided by a mixture of Gaussian *pdfs* and by an HMM.

The real data were fitted to a mixture of normal probability distribution functions before comparing their generated histograms with the histogram for the real data. Table 7 shows the fitted mixtures determined for each WAP. The R package *mixtools* was used to perform the fit. The number of normal distributions for each WAP was set to the maximum number of normal distributions that performed a convergent fit. The probability for each RSSI value in the histogram was calculated by integrating the mixture of Gaussian *pdfs* between $x \in [RSSI_{n-1}, RSSI_n)$.

For each HMM, two histograms were calculated for each WAP. The first histogram was calculated using the probability given by Equation 10 and a length for the sequence of 500 samples. The second histogram was calculated using the probability given by Equation 11 when $t \to \infty$; we name this result "KL-divergence for theoretical HMM".

Table 7: Mixture fits for each WAP. The number of Gaussian *pdfs* in the fit is the optimum number that provides the lowest KL-divergence regarding the real data.

Mac	μ	σ	λ	Mac	μ	σ	λ
1	-79.39057	1.16038	0.07381	4	-67.69326	1.91877	0.11290
	-70.16000	1.14648	0.12401		-62.32896	1.26513	0.42604
	-68.62251	2.27171	0.43916		-58.52591	0.92952	0.18611
	-61.27791	0.70445	0.05236		-56.34613	1.05006	0.27492
	-68.61769	0.64507	0.18644	5	-79.93687	2.23901	0.34554
	-74.27176	1.36520	0.12420		-78.48352	1.33541	0.43515
2	-78.87969	1.29469	0.11639		-73.65112	0.95534	0.21929
	-72.18136	1.35310	0.51508	6	-84.03421	1.48227	0.19544
	-69.39329	0.70383	0.15841		-81.63146	0.67906	0.33126
	-64.57878	1.20245	0.21010		-73.77455	1.03478	0.03534
3	-69.85665	1.63068	0.12973		-79.04889	1.16163	0.43794
	-63.74287	2.21468	0.64044				
	-57.15481	0.98498	0.22981				

KL-divergence (Kullback & Leibler, 1951) was used to compare real and calculated histograms. Results for the KL-divergence are shown in Table 8. In all cases, the KL-divergence between the real histogram and the histograms generated by an HMM are lower than the KL-divergence between the real histogram and the histogram generated by a mixture of Gaussian *pdfs*. Note that for mac_6 the high value of the KL-divergence for the mixture of normal distribution functions is due to the absent values (codified as $RSSI = -100$) that where excluded before the fit. Note also that for all HMM the KL-divergence for the theoretical histogram and the histogram calculated for 500 samples are very similar, with the exception of mac_6. The KL-divergence provided by the theoretical HMM is very similar to the KL-divergence provided by the HMM using 500 samples.

It can be concluded that, for all cases studied, an HMM which fits a real RSSI Wi-Fi signal is able to generate a more accurate histogram than a fit to a mixture of Gaussian *pdfs*.

Table 8: KL-divergence between the histogram from the real sequence of data and the histograms for the HMM of length 500, the histogram for a length tending to infinity, and the optimum mixture of Gaussian *pdfs* to fit the real data.

Mac	KL-div. HMM(500)	KL-div. HMM theoretical	KL-div. Gauss. mixture
mac_1	0.0003	0.0003	0.0510
mac_2	0.0005	0.0005	0.1562
mac_3	0.0139	0.0140	0.0862
mac_4	0.0323	0.0326	0.0587
mac_5	0.0065	0.0065	0.0508
mac_6	0.0027	0.0015	10.4443

6. Conclusions and future work

Wi-Fi fingerprinting uses RSSI Wi-Fi signals to provide location-based services. Although representing RSSI Wi-Fi signals using a Gaussian *pdf* is the preferred model in the domain of Wi-Fi fingerprinting, this work has shown that HMMs are a better model in terms of histogram accuracy and autocorrelation preservation. A well-known characteristic of RSSI Wi-Fi histograms is their being skewed, and this cannot be modeled using only one Gaussian *pdf.* Although using a mixture of Gaussian *pdfs* could be a solution to model the skewed RSSI Wi-Fi histogram, we have shown, using a KL-divergence analysis between the real and modeled histograms, that HMM models RSSI Wi-Fi signals better than a mixture of Gaussian *pdfs.*

Another characteristic present in the RSSI Wi-Fi temporal series is autocorrelation, which cannot be preserved by means of a mixture of Gaussian *pdfs.* Again, HMM can naturally model the autocorrelation present in the RSSI Wi-Fi signal. We have developed the mathematical formulation to calculate the autocorrelation coefficient from the HMM parameters. Through extensive experimentation we have shown the dependency of the calculated autocorrelation coefficients regarding the number of hidden states, and the number of iterations used to create an HMM model. In addition, we have developed the mathematical formulation for calculating autocorrelation coefficients for the static case.

HMM models have two interesting properties that may be used in the domain of Wi-Fi fingerprinting. The first one is that they can be applied to forecast a temporal series in the future, which could be used to create "virtual" WAPs by generating such temporal series. This is one of our lines of future work, to create such "virtual" WAPs in order to improve the accuracy of a Wi-Fi fingerprinting method. The second property is that, like any other generative model, they can offer the probability of generating a sequence of RSSI Wi-Fi samples by the HMM, which could be used to develop a location algorithm. This is another of our lines of future work: to develop an indoor location algorithm which will use the probability of generating a temporal series to estimate the user's location. **Device diversity is a challenging issue when developing indoor location algorithms. The work presented in this paper might be useful to address this issue since RSSI Wi-Fi signal presents autocorrelation regardless the devices used as emitter (WAPs) or receiver (mobile phones, laptops, smart-watches, etc.).**

The main drawback in the use of the HMM to model RSSI Wi-Fi signals is the amount of information needed to store the models compared with a mixture of Gaussian *pdfs*, where only two parameters are needed per Gaussian *pdf*. Nevertheless, we think that the two aforementioned properties of the HMM are very interesting regardless of the overhead in storage space.

Regarding the complexity to use this model in a real scenario, to work with HMMs mostly implies matrix multiplication operations. The size of such a matrix depends on the number of states used in the model. In the experiments presented in this paper, we have used up to 100 states which means a matrix of size 100x100. Although the size of the matrix could seem big, even current hardware present in mobile phones is powerful enough to perform these operations in milliseconds. This operational time could be even reduced if matrix multiplication were performed using graphics hardware already present in most consumer mobile phones. Regarding database cre-

ation time, fingerprinting database creation is made of two stages: i) data collection, and ii) model creation. Data collection is a time consuming task, but it is performed only once. In our case to collect 500 Wi-Fi samples took almost 10 minutes. To create a HMM using the Waum-Welch algorithm depends on two factors: i) the number of samples, and ii) the number of iterations used to fit the model. Although this could be a time consuming task, it is performed only once. In our case, to create a HMM using 500 samples and 100 iterations took almost 20 minutes using an Ideapad laptop equipped with an i7-8750 CPU, 16 GB of RAM and running Ubuntu 20.04.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science, Innovation and Universities through the "Retos investigación" programme (RTI2018-095168-B-C53), by Universitat Jaume I "Pla de promoció de la investigació 2017" program (UJI-B2017-45) and by Generalitat Valenciana "Promoció de la investigació 2020" program (AICO/2020/046).

References

Alletto, S., Cucchiara, R., Del Fiore, G., Mainetti, L., Mighali, V., Patrono, L., & Serra, G. (2015). An indoor location-aware system for an iot-based smart museum. *IEEE Internet of Things Journal*, *3*, 244–253.

Anderson, T. W., & Darling, D. A. (1954). A test of goodness of fit. *Journal of the American statistical association*, *49*, 765–769.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, *41*, 164–171.

Bisio, I., Lavagetto, F., Marchese, M., & Sciarrone, A. (2016). Smart probabilistic fingerprinting for WiFi-based indoor positioning with mobile devices. *Pervasive and Mobile Computing*, *31*, 107–123. URL: `https://www.sciencedirect.com/science/article/pii/S1574119216000377`. doi:`10.1016/J.PMCJ.2016.02.001`.

Bose, A., & Foh, C. H. (2007). A practical path loss model for indoor wifi positioning enhancement. In *2007 6th International Conference on Information, Communications & Signal Processing* (pp. 1–5). IEEE.

Cobos, M., Perez-Solano, J. J., Belmonte, Ó., Ramos, G., & Torres, A. M. (2016). Simultaneous ranging and self-positioning in unsynchronized wireless acoustic sensor networks. *IEEE Transactions on Signal Processing*, *64*, 5993–6004.

Fang, S. H., Lin, T. N., & Lee, K. C. (2008). A novel algorithm for multipath fingerprinting in indoor WLAN environments. *IEEE Transactions on Wireless Communications*, *7*, 3579–3588. URL: `http://ieeexplore.ieee.org/document/4626331/`. doi:`10.1109/TWC.2008.070373`.

Farid, Z., Nordin, R., & Ismail, M. (2013). Recent advances in wireless indoor localization techniques and system. *Journal of Computer Networks and Communications*, *2013*.

Gupta, A., & Dhingra, B. (2012). Stock market prediction using hidden markov models. In *2012 Students Conference on Engineering and Systems* (pp. 1–4). IEEE.

Haeberlen, A., Flannery, E., Ladd, A. M., Rudys, A., Wallach, D. S., & Kavraki, L. E. (2004). Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking* (pp. 70–84). ACM.

He, S., & Chan, S.-H. G. (2015). Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, *18*, 466–490.

Jarque, C. M., & Bera, A. K. (1987). A test for normality of observations and regression residuals. *International Statistical Review/Revue Internationale de Statistique*, (pp. 163–172).

Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.

Kaemarungsi, K., & Krishnamurthy, P. (2004). Properties of indoor received signal strength for wlan location fingerprinting. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.* (pp. 14–23). IEEE.

Kaemarungsi, K., & Krishnamurthy, P. (2012). Analysis of WLAN's received signal strength indication for indoor location fingerprinting. *Pervasive and Mobile Computing*, *8*, 292–316. URL: https://www.sciencedirect.com/science/article/pii/S1574119211001234{#}f000010. doi:10.1016/J.PMCJ.2011.09.003.

Koski, T. (2001). *Hidden Markov models for bioinformatics* volume 2. Springer Science & Business Media.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Ann. Math. Statist.*, *22*, 79–86. URL: https://doi.org/10.1214/aoms/1177729694. doi:10.1214/aoms/1177729694.

Li, Z., Liu, J., Yang, F., Niu, X., Li, L., Wang, Z., Chen, R., Li, Z., Liu, J., Yang, F., Niu, X., Li, L., Wang, Z., & Chen, R. (2018). A Bayesian Density Model Based Radio Signal Fingerprinting Positioning Method for Enhanced Usability. *Sensors*, *18*, 4063. URL: http://www.mdpi.com/1424-8220/18/11/4063. doi:10.3390/s18114063.

Molina, B., Olivares, E., Palau, C. E., & Esteve, M. (2018). A multimodal fingerprint-based indoor positioning system for airports. *IEEE Access*, *6*, 10092–10106.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Pearson, K. (1900). X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *50*, 157–175.

Pritt, N. (2013). Indoor location with wi-fi fingerprinting. In *2013 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)* (pp. 1–8). IEEE.

Pu, Y.-C., & You, P.-C. (2018). Indoor positioning system based on ble location fingerprinting with classification approach. *Applied Mathematical Modelling*, *62*, 654 – 663. URL: `http://www.sciencedirect.com/science/article/pii/S0307904X18302841`. doi:`https://doi.org/10.1016/j.apm.2018.06.031`.

Qi, C., Gaoming, H., & Shiqiong, S. (2009). WLAN user location estimation based on receiving signal strength indicator. In *Proceedings - 5th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2009* (pp. 1–4). IEEE. URL: `http://ieeexplore.ieee.org/document/5305128/`. doi:`10.1109/WICOM.2009.5305128`.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*, 257–286. doi:`10.1109/5.18626`.

Santoso, F., & Redmond, S. J. (2015). Indoor location-aware medical systems for smart homecare and telehealth monitoring: state-of-the-art. *Physiological measurement*, *36*, R53.

Seneta, E. (2006). *Non-negative matrices and Markov chains*. Springer Science & Business Media.

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*, 591–611.

Smailagic, A., Small, J., & Siewiorek, D. P. (2000). Determining user location for context aware computing through the use of a wireless lan infrastructure. *Institute for Complex Engineered Systems Carnegie Mellon University, Pittsburgh, PA*, *15213*.

Torres-Sospedra, J., Montoliu, R., Trilles, S., Belmonte, Ó., & Huerta, J. (2015). Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems. *Expert Systems with Applications*, *42*, 9263–9278.

Werner, M. (2014). *Indoor location-based services: Prerequisites and foundations*. Springer.

Yang, C., & Shao, H.-R. (2015). Wifi-based indoor positioning. *IEEE Communications Magazine*, *53*, 150–157.

Youssef, M., & Agrawala, A. (2008). The horus location determination system. *Wireless Networks*, *14*, 357–374.

Zucchini, W., MacDonald, I. L., & Langrock, R. (2017). *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC.

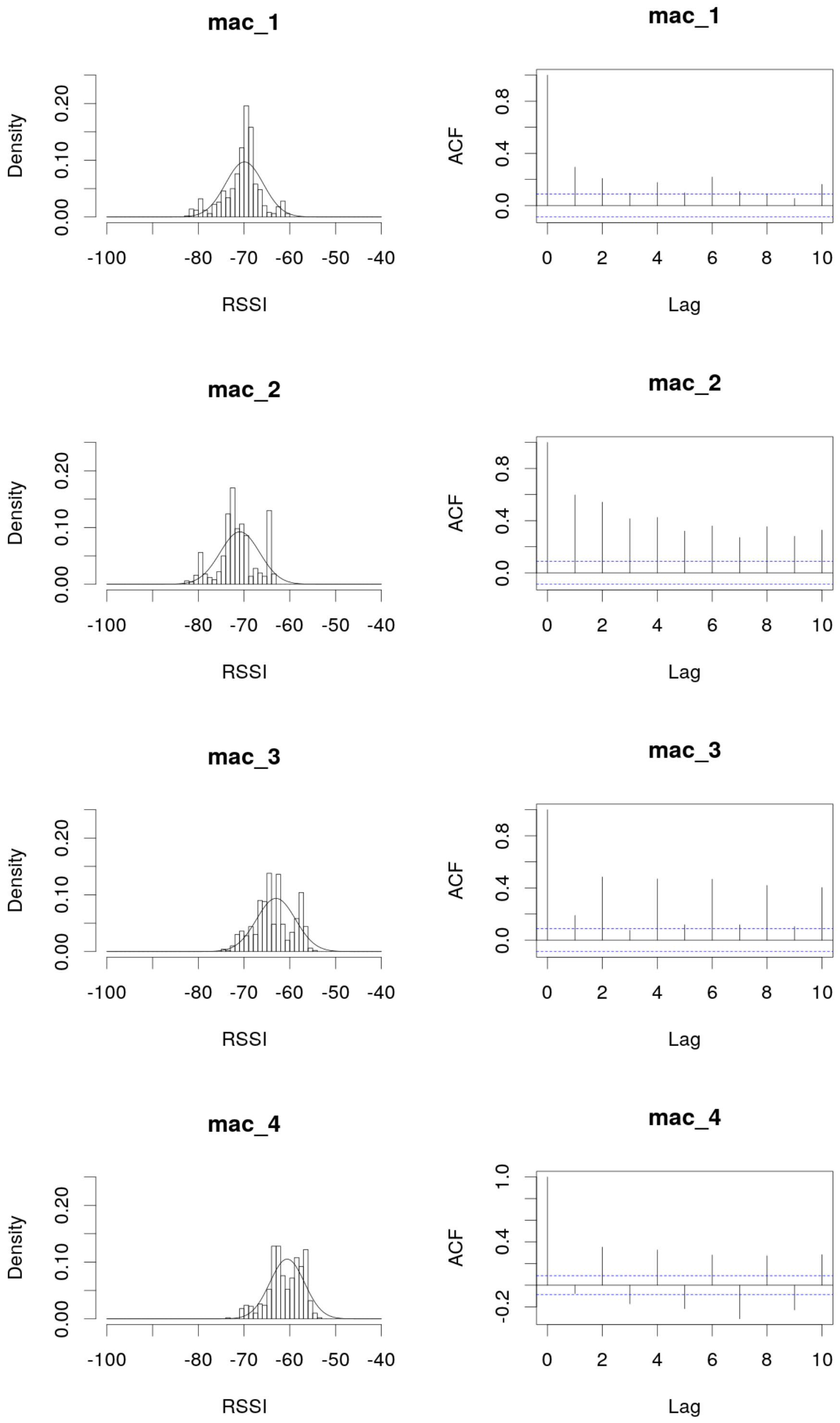


Figure 3: The histograms for mac_1 to mac_4 using 500 samples are shown on the left. The first ten autocorrelation coefficients are shown in plots on the right; the horizontal dotted lines represent the value 0.1 (top) and the value -0.1 (bottom).

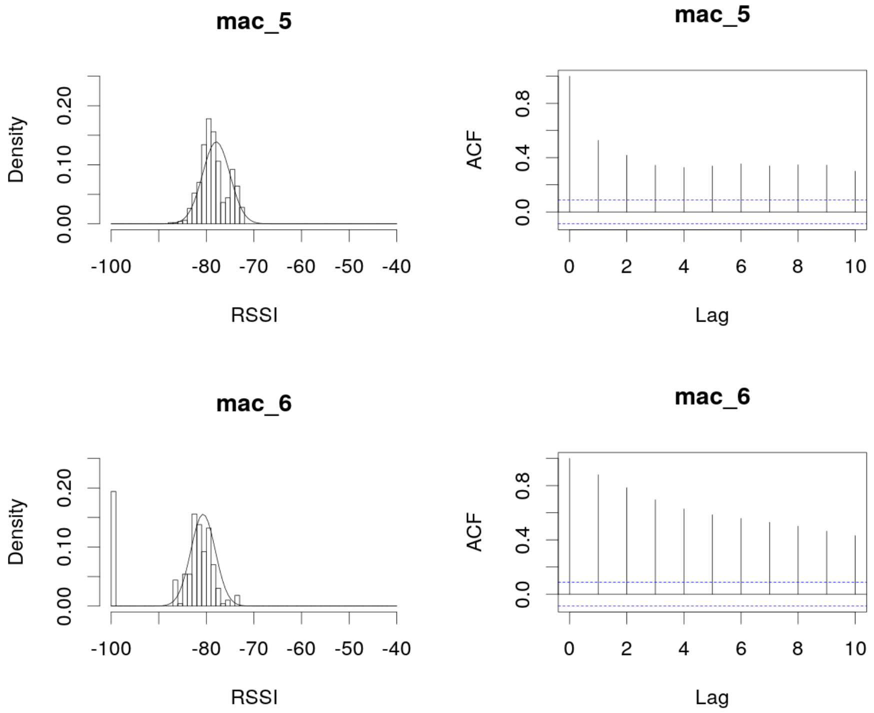


Figure 4: The histograms for mac_5 amd mac_6 using 500 samples are shown on the left. The first ten autocorrelation coefficients are shown in plots on the right; the horizontal dotted lines represent the value 0.1 (top) and the value -0.1 (bottom).

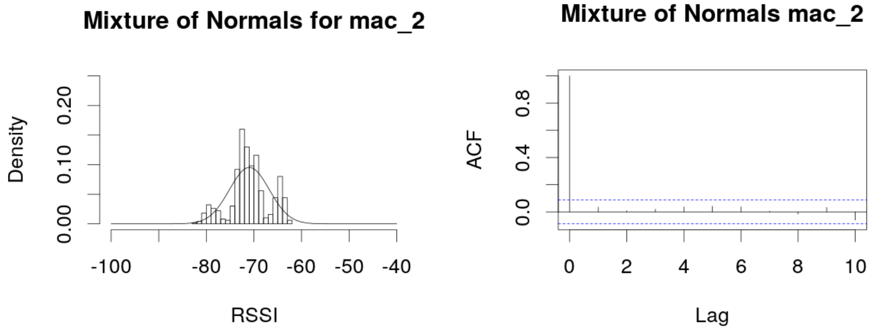
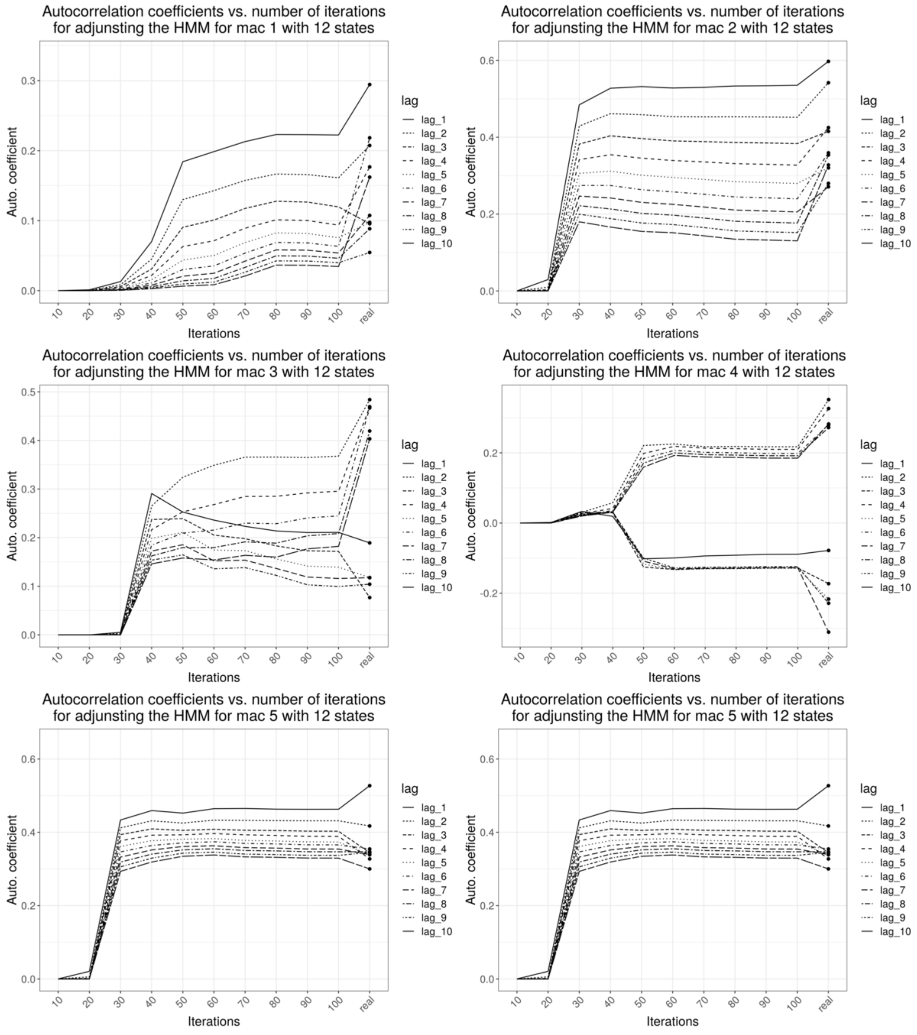


Figure 5: Data histogram and autocorrelation coefficients for real RSSI Wi-Fi signals for mac_2, and for the fit of a mixture of normal distributions for mac_2. The superimposed curve in the histogram figure is for the Gaussian *pdf* fit of the data.



Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 1 with 12 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 2 with 12 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 3 with 12 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 4 with 12 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 5 with 12 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 5 with 12 states

Figure 6: Accuracy of autocorrelation coefficients regarding the number of iterations to create the HMM. The number of states was 12 for all the HMMs created. The data in the last row of the x-axis (solid circles) is for the autocorrelation values calculated from the real data.

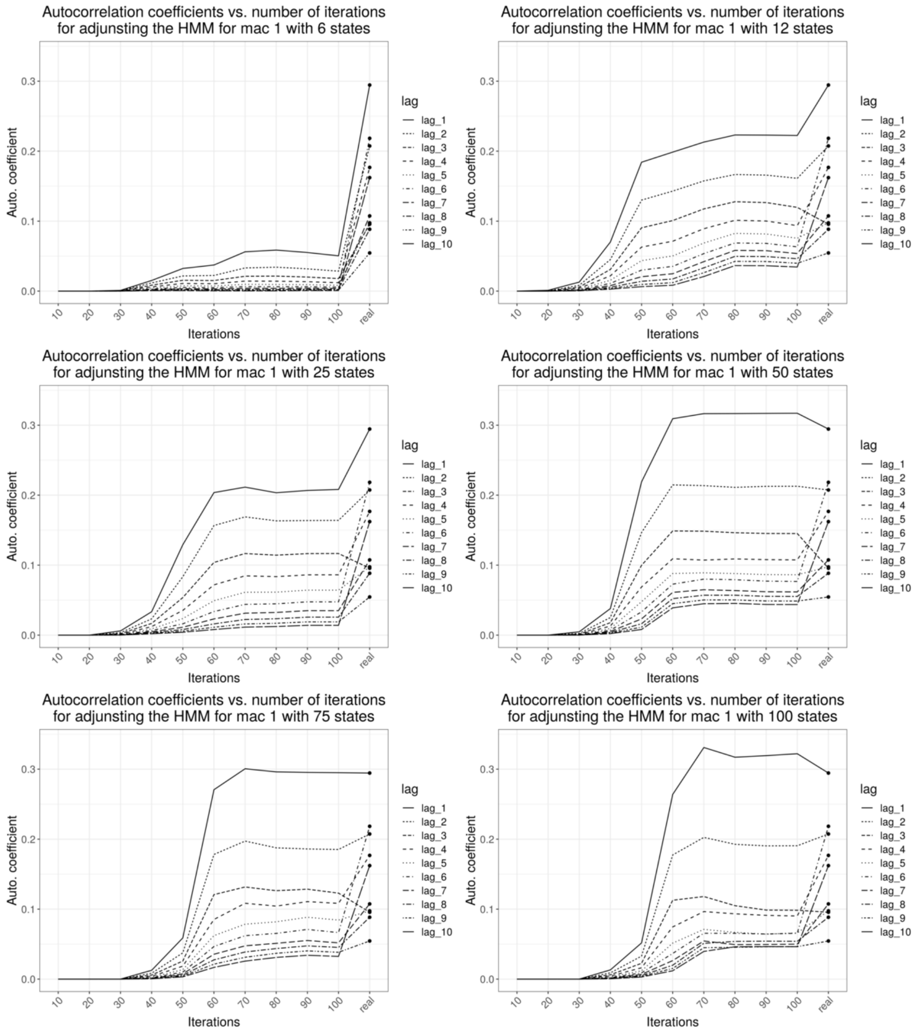
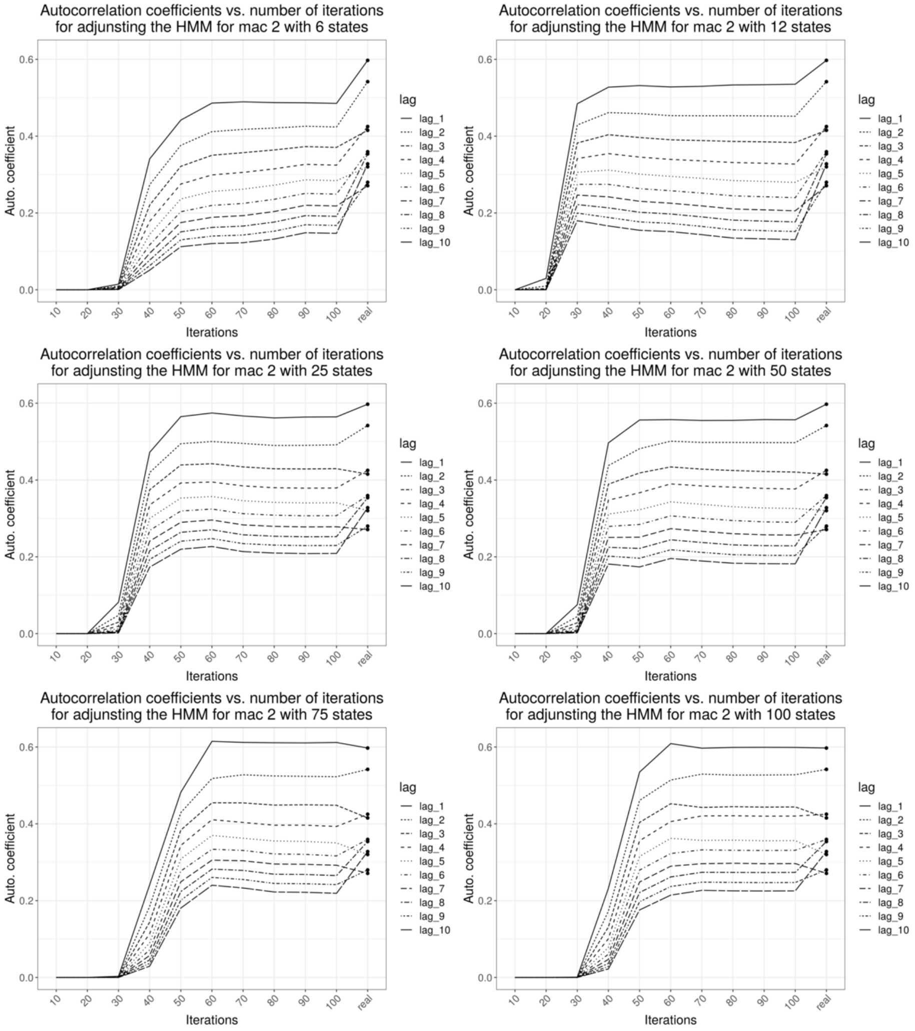


Figure 7: Autocorrelation coefficients when the number of hidden states used to create the HMM was 6, 12, 25, 50, 75 and 100. Each plot shows the evolution, for a fixed number of hidden states, when the number of iterations ranges between 10 and 100 in steps of 10.


Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 2 with 6 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 2 with 12 states

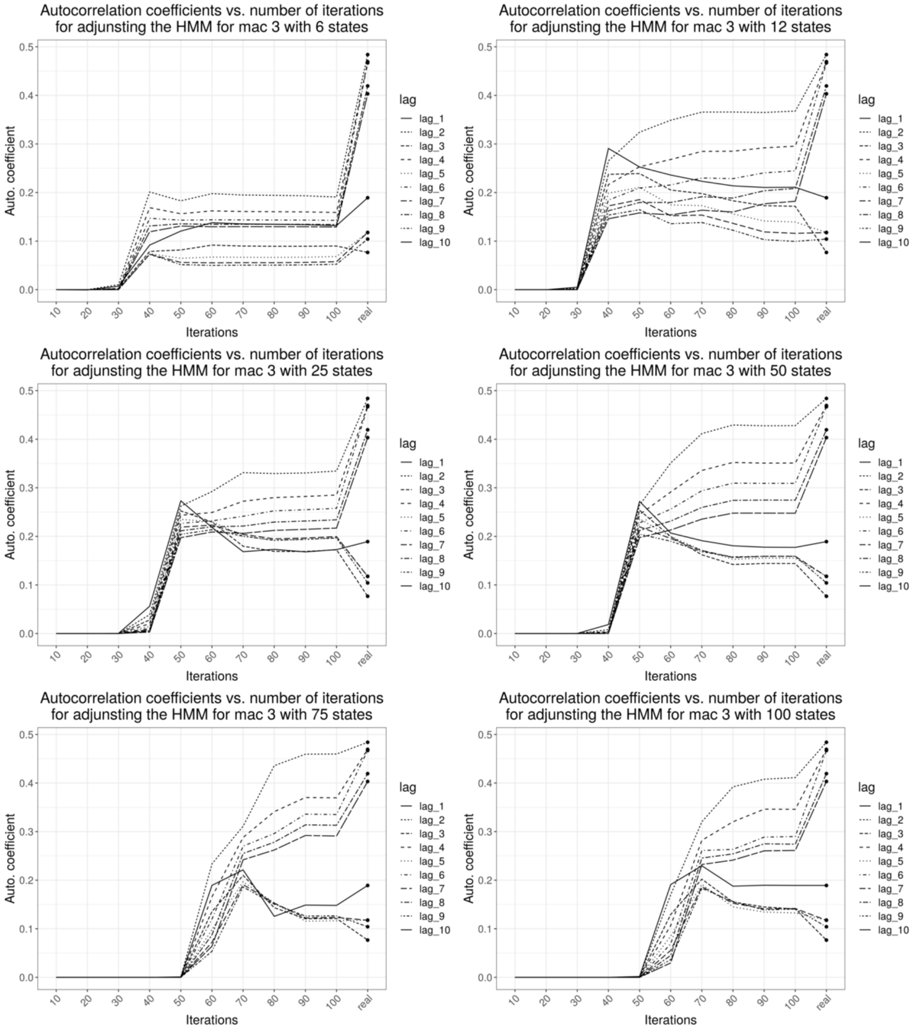
Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 2 with 25 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 2 with 50 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 2 with 75 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 2 with 100 states

Figure 8: Autocorrelation coefficients when the number of hidden states used to create the HMM was 6, 12, 25, 50, 75 and 100. Each plot shows the evolution, for a fixed number of hidden states, when the number of iterations ranges between 10 and 100 in steps of 10.


Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 3 with 6 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 3 with 12 states

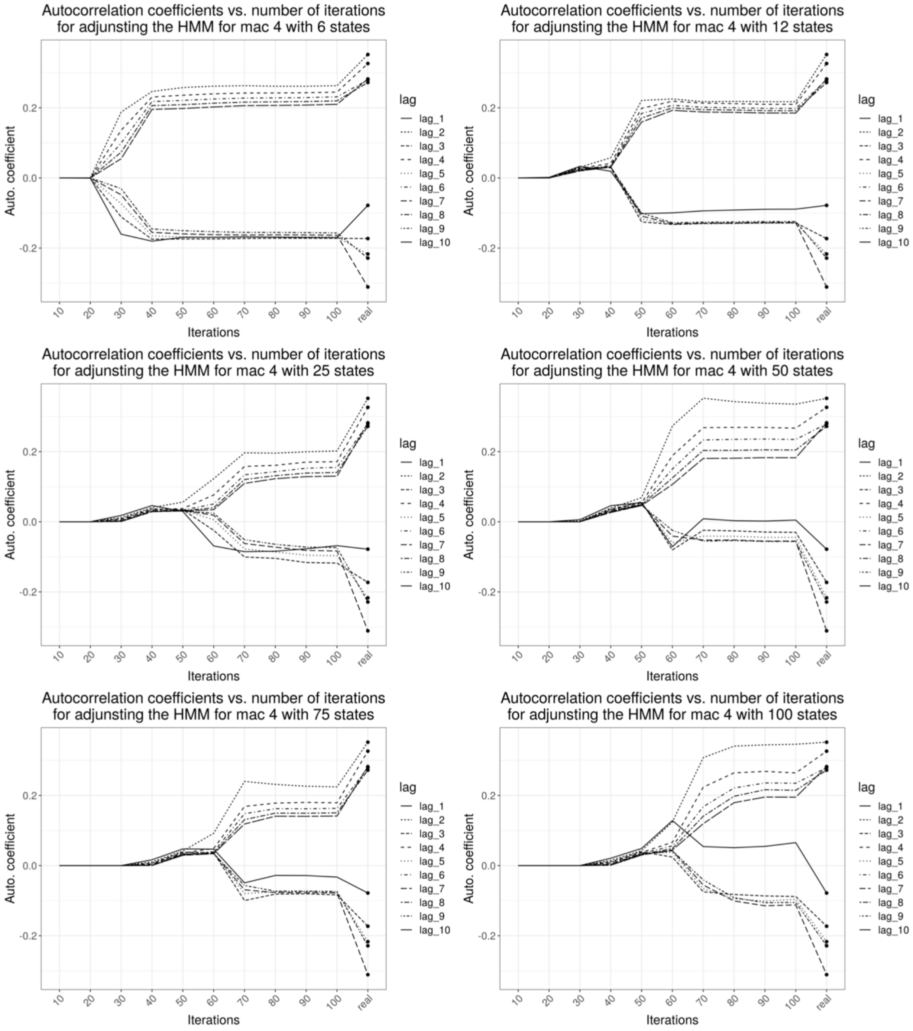
Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 3 with 25 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 3 with 50 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 3 with 75 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 3 with 100 states

Figure 9: Autocorrelation coefficients when the number of hidden states used to create the HMM was 6, 12, 25, 50, 75 and 100. Each plot shows the evolution, for a fixed number of hidden states, when the number of iterations ranges between 10 and 100 in steps of 10.



Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 4 with 6 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 4 with 12 states

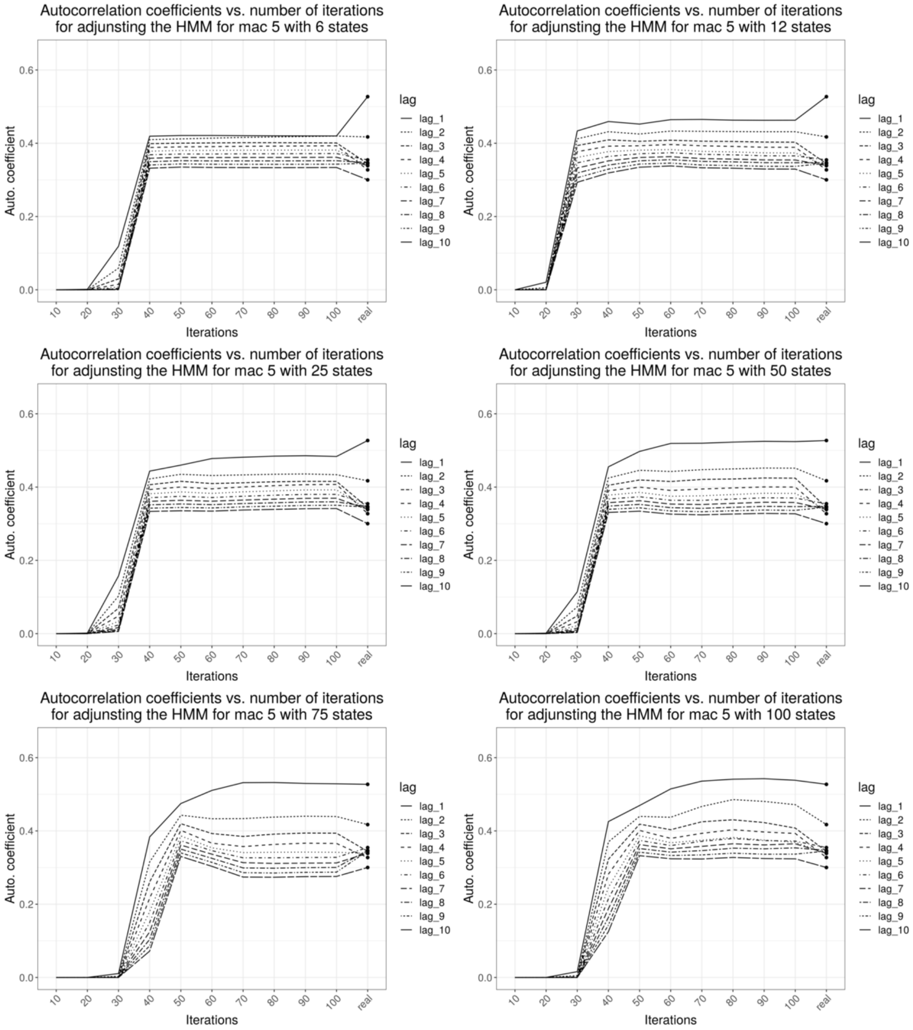
Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 4 with 25 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 4 with 50 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 4 with 75 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 4 with 100 states

Figure 10: Autocorrelation coefficients when the number of hidden states used to create the HMM was 6, 12, 25, 50, 75 and 100. Each plot shows the evolution, for a fixed number of hidden states, when the number of iterations ranges between 10 and 100 in steps of 10.


Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 5 with 6 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 5 with 12 states

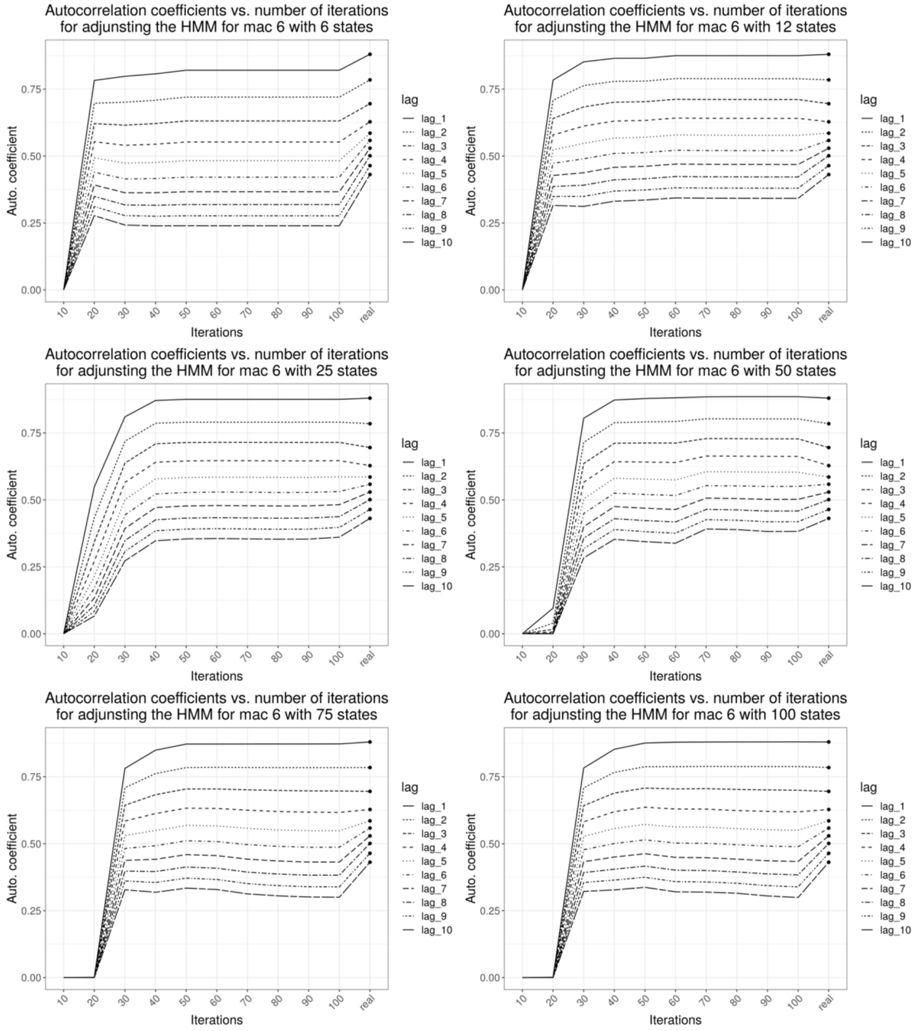
Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 5 with 25 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 5 with 50 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 5 with 75 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 5 with 100 states

Figure 11: Autocorrelation coefficients when the number of hidden states used to create the HMM was 6, 12, 25, 50, 75 and 100. Each plot shows the evolution, for a fixed number of hidden states, when the number of iterations ranges between 10 and 100 in steps of 10.



Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 6 with 6 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 6 with 12 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 6 with 25 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 6 with 50 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 6 with 75 states

Autocorrelation coefficients vs. number of iterations for adjunsting the HMM for mac 6 with 100 states

Figure 12: Autocorrelation coefficients when the number of hidden states used to create the HMM was 6, 12, 25, 50, 75 and 100. Each plot shows the evolution, for a fixed number of hidden states, when the number of iterations ranges between 10 and 100 by 10 steps.