Wifi tracking

You are going to track a smart phone through an employee restaurant using the phone's wifi signals. The restaurant is at the KPMG head quarters at Amstelveen and the data that you will use are actual wifi signal data from one particular smart phone. The phone's owner has given consent on using his data.

To do wifi tracking the KPMG Big Data team has built a system that measures wifi signals from wifi enabled devices and has developed algorithms that reconstruct the position of the devices from the measured signals. In this exercise you yourself will also develop such an algorithm. The system consists of some eleven normal wifi routers that have been reprogrammed to function as wifi sensors. This basically means that the routers no longer pass through information to and from the internet, but only listen to and record all wifi traffic around them.

0.1 Wifi signals

Wifi communication is done in wifi packets whose format follows the 802.11 protocol. For tracking purposes the packet content (e.g. an http request) is not so interesting, but rather the packet header is, which contains among others

- The wifi MAC address of the device the packet is sent from. The MAC address is generally hashed and/or replaced with a pseudonym in the tracking system out of privacy concerns.
- The type and subtype of the packet indicating the packet's purpose. A packet can convey data/content like an http request, but can also be a management packet between phone and network conveying information on how to keep a stable connection. For tracking purposes the type and subtype are not so interesting, apart from the fact that they are needed to uniquely identify a packet.
- The sequence number of the packet. Every packet of every type and subtype is sent with a sequence number, which is thereafter incremented before a next packet of the type and subtype is sent. After running through 4096 values (12 bits) the sequence number loops back on itself.

There are other header items that can be important such as the retry flag, the fragment number and destination MAC address, but for convenience the subtleties of these have been taken out of the data set. When the router registers a wifi packet the header information is supplemented with among others

- A time stamp when the packet is registered, in Unix time with millisecond precision.
- The signal strength of the packet as seen by the router, in decibels with respect to 1 mWatt (dBm).
- The name of the router that registered the wifi packet.

0.2 Signal strength and distance

The basic idea behind determining the location of a wifi device is that the signal strength of a wifi packet is larger for routers that are closer to the device than for routers that are farther away. Thus the position of the device is somehow set in the different signal strengths of the same wifi packet at different routers. The relation between signal strength at the router and the distance between device and router is given by the Friis free space equation.

$$P_r = P_t + 20 \times {}^{10}\log\left(\frac{c}{4\pi f r}\right) \tag{1}$$

Where P_t is the transmission powe (the original signal strength of the packet at the phone), where $c = 299792458 m/s$ is the speed of light, $f = 2.4 GHz$ is the signal frequency, and r is the distance between device and router. The transmission power P_t and the signal strength at the router (receiver power) P_r are in this equation measured in dBm, decibels with respect to 1 mWatt. The transmission power is generally also considered an unknown, since it may vary from phone to phone, from packet type to packet type, and due to for example power saving modes.

a. Make a plot of P_r as a function of r when $P_t = 0\ dBm$, for r ranging from $0.4\ m$ to $30\ m$[1]. For which is a router more sensitive when it comes to the distance to a device, devices that are close or devices that are far away, and why?

The Friis free space equation 1 relates the signal strength of a packet at the router to the distance between router and device. However, in reality you will hardly ever find this relation to hold exactly, because of numerous noise and other influences. For example the chip, circuitry and antenna of the router can pick up electronic noise, or other electromagnetic sources may interfere with the signal. These and more influences will make that the actual measured signal strength will likely not be exactly the same as dictated by the Friis equation 1, but fluctuate randomly from it.

b. Invert the Friis equation 1 to give the distance r as a function of P_r.

c. When $P_t = 0\ dBm$, what is the difference in distance between a signal strength of $-30\ dBm$ and of $-31\ dBm$, and between a signal strength of $-60\ dBm$ and $-61\ dBm$?

d. For which case does an uncertainty on the measured signal strength translate to a larger uncertainty on the distance, for a larger signal strength of e.g. $-30\ dBm$ or for a smaller signal strength of e.g. $-60\ dBm$, and why?

[1] The Friis free space equation is an approximation that brakes down when the distance between transmitter and receiver is too small, less than approximately 40 cm.

0.3 Position reconstruction

The Friis equation 1 in principle lets you draw a circle around every router indicating the distance from which the packet was sent. Then the device's location is at the intersection of all the circles. Even with the transmission power unknown, you can in principle vary the transmission power, which acts as a scale factor to the size of the circles, until the circles intersect in one point. In practice however, due to noise and other influences, you will not find an exact intersection point. In stead we will look for a location which is 'closest' or 'most likely' to being the intersection point.

e. Explain and draw an example of why there will not be an exact intersection point?

If the wifi device were at a position $[x, y]$ and transmits with a transmission power P_t, then the expected signal strength for router i at position $[x_i, y_i]$ is

$$P_r^i(x,y) = P_t + 20 \times {}^{10}\log\left(\frac{c}{4\pi f}\right) - 10 \times {}^{10}\log\left((x-x_i)^2 + (y-y_i)^2 + Z^2\right)$$

Where Z is the difference in height between device and router. Since phones are mostly kept at pocket height we don't keep the z-coordinate as a variable but fix Z at a value of $Z = 2\ m$ (the routers are placed approximately $3\ m$ up and a pants pocket is approximately $1\ m$ up).

f. Derive the above equation from the Friis equation 1.

Of course we don't know the position $[x, y]$ of the device, estimating it is the whole purpose here.

Suppose the router measures a signal strength S_i. The difference between measured and expected $S_i - P_r^i(x,y)$ is called the *residual.* Due to noise we don't expect the residual to be zero, even if $[x, y]$ is the true position of the device. Suppose that we estimate that due to noise there can be a variation/measurement uncertainty of σ_i in the measured signal strength at router i. In general the measurement error can be different from packet to packet, router to router, and even depend on the distance or other factors. It is up to the analyst to give a proper estimate of this measurement error.

The normalized residual $\frac{S_i - P_r^i(x,y)}{\sigma_i}$ expresses how close the measured and expected signal strength are to each other if we assume the device to be at position $[x, y]$. It expresses it in how many times the uncertainty it is off, how many *standard deviations* from zero. The larger the normalized residual is, the more likely it is that the difference between expectation and measurement is the result of wrong parameters than of noise. In this case the more likely it is that the difference $S_i - P_r^i(x,y)$ between expected and measured signal strength is the result of the estimated position $[x, y]$ being wrong than from measurement errors.

g. Make a small simulation, where you have one router at coordinate $[0, 0]$, a mobile device at coordinate $[20, 0]$ which transmits with power $P_t = 0\ dBm$. The router measures wifi packets from the mobile device according to the Friis equation 1, but the signal strength at the router obtains an additional random Gaussian fluctuations of $1\ dBm$ around the Friis equation value. Generate 1000 such packets (incl. fluctuations). For each packet calculate the expected signal strength (this is just the Friis equation value) and calculate the normalized residual when you estimate the measurement uncertainties (correctly) at $\sigma = 1$. Plot the distribution of the normalized residuals. This distribution is called a *pull distribution.* What is the mean and the standard deviation of the pull distribution?

h. Now generate another 1000 packets but with 2 dBm random Gaussian fluctuations. However, in determining the normalized residuals keep $\sigma = 1$ dBm in the denominator. This is the case when in truth the noise has an amplitude of 2 dBm while you underestimate it to have an amplitude of 1 dBm. What happens to the pull distribution if you underestimate the noise/fluctuations/measurement uncertainties? And what will happen if you overestimate?

i. Now generate another 1000 packets again with 1 dBm fluctuations, but this time replace the 20 in the Friis equation 1 by 25. However, in determining the expected signal strength for the normalized residuals keep the Friis equation as it is. The factor 20 in the Friis equation may in reality be slightly different due to signal reflections, or signal attenuation by obstacles, or non-spherical radio wave fronts. In this case you have a systematic difference between your model and the data, rather than random fluctuations. What happens to the pull distribution if you have such a systematic error in your model?

j. Pull distributions, if you can make them, are a powerful way to judge if your model agrees with the data. In general, if for any problem, data analysis or other experiment that you may encounter in your life you can make a pull distribution, what do you want to see in your pull distribution to convince you that your model is correct and your uncertainties properly estimated?

If we square the normalized residual and sum them up for all routers

$$\chi^2\left(x,y\right) = \sum_i \frac{\left(S_i - P_r^i(x,y)\right)^2}{\sigma_i^2}, \tag{2}$$

then we obtain a Euclidean distance measure that expresses how close the measured signal strengths are to the expected signal strengths if the device is assumed to be at position $[x, y]$. Such a distance measure of a sum of squared normalized residuals is a called a *chi-squared.* And we say that the position $[x, y]$ which has the smallest chi-squared is closest to being the intersection point and most likely the device's true position.

0.4 Toy Monte Carlo

Before we work on actual data we are going to build a simulation. We often call the type of simulation you will build a *Toy Monte Carlo simulation.* In such a simulation you build a simplified (toy) world to understand how different factors can influence your real complex system. Often you start by first building a world that agrees exactly with your model and then adding disruptions to see what happens.

Suppose we have four routers at positions $[0, 0]$, $[0, 20]$, $[20, 20]$ and $[20, 0]$, and they are all 3 m up high. Suppose furthermore that we have a wifi enabled device at position $[5, 5]$, at pocket height (1 m up high), that sends out wifi packets with transmission power $P_t = 0\ dBm$

k. Generate a single wifi packet, determine the signal strength it produces at each of the four routers (without noise). Then make a plot of the χ^2 as a function of assumed x-position for x between $-5\ m$ and 15 m and $y = 5\ m$. For the χ^2 calculation let your estimated measurement uncertainty be $\sigma_i = 1\ dBm$ for each of the four routers. The χ^2 should be minimum at the true position $[5, 5]$.

l. Again generate a single wifi packet, and determine the signal strength it produces at each of the four router, but this time add random Gaussian fluctuations of 1 dBm to the measured signal strength at each of the four routers independently. Again make a plot of the χ^2 as a function of assumed x-position for x between $-5\ m$ and 15 m and $y = 5\ m$. Does the minimum occur at the true position $[5, 5]$? And what happens to the minimum if you repeat this procedure several times (with different independent random fluctuations)?

m. The χ^2 has a minimum at some point $[x, y]$. Use a minimization procedure (for python you can use scipy.optimize.minimize) to find the $[x, y]$ for which the χ^2 is minimum.

n. Generate 1000 wifi packets (each one with random Gaussian fluctuations at the routers). For each one find the $[x, y]$ position that minimizes the χ^2 and the χ^2 at this minimum. Plot the positions in a scatter plot. Plot the χ^2 minimums in a histogram. What is the average x- and y-position? What is the average χ^2 of the minimums?

In an experiment or a simulation as you just did, you expect on average to find a χ^2 equal to the *number of degrees of freedom.* The number of degrees of freedom, or *NDoF*, is the number of data points that you have minus the number of parameters that you are estimating. In this case you have four data points, the four signal strengths measured at each of the four drones. And you have two parameters that you are estimating, the x- and y-position of the wifi device. Thus in this case you have two degrees of freedom. The idea behind the NDoF is that not all your data points are 'free', but a number of them 'are needed' to fix your parameters. The rest of them are then free to deviate from what you expect and crank up your χ^2.

In fact you expect your minimums to be distributed according to a chi-squared distribution. Go online and find a text on chi-squared distributions, e.g. Wikipedia, just to have a notion on what the distribution looks like for different numbers of degrees of freedom.

o. Does the average χ^2 of the minimums that you found in the previous item agree with the NDoF?

p. Use a library to plot the chi-squared distribution with two degrees of freedom on top of the distribution of your minimums (for Python you can use scipy.stats.chi2.pdf). You may have to normalize your histogram of minimums. Does your minimums histogram agree with the chi-squared distribution?

q. Again generate 1000 wifi packets but now with random Gaussian fluctuations of 2 dBm. For each one find the $[x, y]$ position that minimizes the χ^2 and the χ^2 at this minimum, but still assuming a measurement uncertainty of $\sigma_i = 1$ dBm at each of the routers. Plot the positions in a scatter plot. Plot the χ^2 minimums in a histogram (normalized if you need). Plot the chi-squared distribution with two degrees of freedom on top of the histograms of minimums. What is the average x- and y-position? What happened to the cloud of estimated device positions, and why? What is the average χ^2 of the minimums? Does it agree with what you expect? Does your histogram agree with what you expect from a chi-squared distribution? What happened and why?

In the last exercise you simulated the important case where you as an analyst underestimate the measurement uncertainty. In reality you had random fluctuations of 2 dBm, while you estimated them to be 1 dBm, and you saw this in your average χ^2 value (and in the distribution).

r. Suppose you repeatedly do a chi-squared fit where you have 10 degrees of freedom, and you find your average minimum chi-squared to be 40.0. What does this tell you about your estimate of the measurement/data uncertainties?

In some cases you do not even have an estimate of measurement uncertainties. Then you can opt to leave them out, technically setting them to $\sigma_i := 1$ in equation 2. In this case you are then performing a *least squares* fit rather than a chi-squared fit. You can still get a notion of your measurement error by scaling your σ_i's up or down from 1 until your average χ^2 equals the number of degrees of freedom. However, keep in mind that this is just a notion. In reality different data points may have different uncertainties, i.e. σ_i may not be the same for all i. You as an analyst should always try to understand how your data is influenced.

0.5 Error on fit parameters

In the foregoing exercises you had 1000 identical wifi packets being sent from the same device position. Due to noise at the routers you saw that the reconstructed device position deviates from the true device position. But the average of all 1000 reconstructed positions agrees well with the true device position.

In reality you will not have 1000 identical wifi packets being sent from the same device position, you will mostly have just one. Thus you don't have the luxury of averaging over 1000 reconstructed positions, you will just have a single estimate, and this estimate as you have seen can be quite off due to measurement uncertainties. Thus your measurement uncertainties translate into an uncertainty on your reconstructed device position. And as a good academic and data scientist you do not only provide your best estimate of the device location, but also the uncertainty on its location!

There is debate on how to properly calculate the uncertainty on fit parameters (in this case the device position), but in case you just want a fairly rough estimate of how far from the true position your reconstructed position can be off, it can be done fairly easily by determining the *covariance matrix* of the parameters.

When you do a chi-squared fit you find the parameters (here the device x- and y-position) which minimize the chi-squared. Call this their optimal values. If you then change your parameters slightly from their optimal values, then of course your chi-squared will increase. Now the variation in your fit parameters from their optimal value that raises the chi-squared by 1 is then a measure for the uncertainty on your fit parameters.

You can do this search for the variations programmatic by 'scanning' the chi-squared value around the minimum, but when you have a lot of fit parameters this becomes quite involved. In stead, you can get an approximation by linearizing your model around the optimal values and calculate the necessary variations analytically.

A linearization of our model around the optimal position, call it $[x_0, y_0]$, is given by

$$P_r^i(x,y) \approx P_r^i(x_0,y_0) + \frac{\partial P_r^i(x_0,y_0)}{\partial x} \times (x - x_0) + \frac{\partial P_r^i(x_0,y_0)}{\partial y} \times (y - y_0) \quad (3)$$

This is a Taylor expansion of our Friis equation to first order around the optimal value $[x_0, y_0]$.

s. Fill this approximation of our model in the chi-squared equation 2 to obtain an approximation of the chi^2 in the neighborhood of its minimal value $[x_0, y_0]$ (you don't have to work out P_r^i or $\frac{\partial P_r^i}{\partial x}$ yet, just leave them as symbols). The result should be a quadratic equation in $(x - x_0)$ and $(y - y_0)$.

From chi-squared minimization we know that the optimal position $[x_0, y_0]$ is actually the one that solves the two equations

$$\frac{\partial \chi^2}{\partial x} = 2\sum_i \frac{S_i - P_r^i(x_0,y_0)}{\sigma_i^2} \times \frac{\partial P_r^i(x_0,y_0)}{\partial x} = 0 \quad (4)$$

$$\frac{\partial \chi^2}{\partial y} = 2\sum_i \frac{S_i - P_r^i(x_0,y_0)}{\sigma_i^2} \times \frac{\partial P_r^i(x_0,y_0)}{\partial y} = 0 \quad (5)$$

t. Use the two equation 4 and 5 in your previous approximation and derive an equation for the χ^2 in terms of only constant and quadratic terms in $(x - x_0)^2$, $(y - y_0)^2$ and $(x - x_0)(y - y_0)$, no linear terms in $(x - x_0)$ or $(y - y_0)$.

u. For convenience now call $(x - x_0) := \Delta x$ and $(y - y_0) := \Delta y$. You can write your approximation as a matrix equation $\chi^2(\Delta x, \Delta y) \approx A + [\Delta x, \Delta y] \cdot \hat{B} \cdot [\Delta x, \Delta y]$. What is the constant A and what are the matrix elements of $\hat{B}$ (symbolically)?

Now the diagonal elements of the matrix that you just built are directly related to the uncertainties Δx and Δy on the device position, they are in fact the inverse of their *variance*.

v. How large does Δx maximally have to be to raise the χ^2 by 1 ~~if you keep $\Delta y = 0$~~. This is the variance on the x-position estimate. And equally how large does Δy have to be to raise the χ^2 by ~~1 if $\Delta x = 0$.~~ This is the variance on the y-position estimate.

The off-diagonal elements of the matrix are directly related to the correlation between the two coordinates of the device position, they are the inverse of their *covariance.*

w. Generate again a single wifi packet with random Gaussian fluctuations of 1 dBm at each of the routers. Use the minimization procedure to estimate the device's location assuming a (correct) measurement uncertainty of 1 dBm. Now also calculate the two diagonal elements of the matrix $\hat{B}$ that you previously derived (differentiate the Friis equation 1 with respect to x and y on paper and program the result in a calculation of the elements of $\hat{B}$). Use these two diagonal elements to determine the variance of the x- and y-position of the device, and then the standard deviations (square root of the variance). Plot the estimated position of the device and draw around it an ellipse whose width is equal to the standard deviation in x and whose height is equal to the standard deviation in y. The ellipse represents your uncertainty on the estimated device position. Also plot the true location of the device for comparison.

x. You can repeat the above procedure for a few wifi packets if you like.

y. Generate 1000 wifi packets with random Gaussian fluctuations of 1 dBm at each of the routers. For each of the wifi packets estimate the device's location and uncertainty on the location, assuming a (correct) measurement error of 1 dBm.Make pull distributions of the estimated x- and y-positions. Are the pull distributions centered where you expect, and are they as wide as you expect? Could you think of reason(s) why the pull distributions would be (slightly) different from what you expect?

z. What would happen to your estimated errors on the x and y positions if you underestimate your measurement error by a factor 2?

Finally a bonus question
In reality the error ellipses are not horizontal or vertical as you just plotted them, this is just an approximation (on top of one anyway). In reality they are generally diagonally oriented and could even be skewed (the two axes of the ellipse not perpendicular). Derive the proper form the ellipses from the matrix $\hat{B}$ that we derived previously. Why is the ellipse never skewed?

0.6 Wifi tracking

In the next section you will use the foregoing to actually track a smart phone through a restaurant. The restaurant is the employee restaurant at the KPMG head quarters at Amstelveen, and the phone's owner has given consent on using his data.

You are given a csv file containing the measurements of the wifi signals that were sent out by the phone. Every line in the file is a wifi packet registered

column name	description	unit
sourceMac	The hashed MAC address the wifi packet was sent from	
typeNr	Number indicating the type of the packet	0,1, or 2
subTypeNr	Number indicating the sub type of the packet	0,...,11
seqNr	Number indicating which in a sequence of sent packets of this type and sub type this is	0,...,4096
retryFlag	Flag indicating whether this packet has been sent before or not	0,1
measurement Timestamp	Timestamp when the packet was registered	UTC in ms precision
droneId	Name of the router that registered the packet	
signal	Signal strength of the packet as measured by the router	dBm

Table 1: Description of the columns in the csv data file.

by one of the routers. A description of the columns contained in the csv file is given in table 1. Further information about how wifi packets are sent was given at the start of this document.

router name (droneId)	location x,y
Lima	5.82, 5.48
Mike	11.33, 9.43
Kilo	12.39, 6.77
Oscar	2.48, 7.36
Alpha	8.53, 2.16
India	2.18, 5.61
November	8.34, 4.13
Hotel	5.43, 4.71
Romeo	[10.99, 5.94]
Quebec	[6.82, 9.78]
Papa	[9.9, 10.39]

Table 2: Location of the routers.

To track the smart phone you will also need the locations of the routers. These are given to you in table 2. The x- And y-positions are in meters with respect to some irrelevant origin. The routers are all placed 3 m up high and you may assume the smart phone to be kept at pocket height of 1 m up.

0.6.1 Your goal

In the previous exercise you have been taken to the process of setting up a chi-squared fit step by step. In this exercise you will use these steps and their implications to track a smart phone from real data. Your goal it to perform all of the following:

a. Identify the various wifi packets that were sent.

b. Estimate the device's position for each one of the wifi packets.

c. Estimate the uncertainty on the device's position for each one of the wifi packets.

d. Plot the device's positions and uncertainties together with the router locations.

e. Estimate our system's resolution, i.e. how well can we determine a device's location.

f. Estimate the average transmission power (in dBm) of the device.

g. Some wifi packets were sent so shortly after each other that the device could not have moved very far if at all. Compose another chi-squared method to combine the locations of wifi packets that are closely spaced in time into one location.

h. Plot the device's positions and uncertainties from the combination of closely spaced packets together with the router locations.

Document your analysis well, e.g. with inline explanations or additional outside documentation. Make sure your reader can understand what you did, imagine for example that your analysis is a publication going to be peer reviewed. **Plots (histograms, scatter plots) help you understand the data, and help us understand how you did this exercise.**

0.6.2 Some hints and help

The transmission power P_t is not known, so it is a parameter of the chi-squared method.

Wifi packets are uniquely identified by their source MAC address, type, sub type and sequence number, but keep in mind that a sequence number loops back on itself and may not be unique anymore after some time[2]. Wifi signals travel at light speed, so differences in distance to the different routers will not cause any discernible time difference between the same packet at different routers. However, due to queuing and processing times of the router hardware there may be slight differences in measurement times.

We actually do not know what the uncertainties on the measured signal strengths are at the routers. They are at least 1 dBm, because the hardware only gives values without a decimal point. However with signal attenuation, reflection, interference from other sources, and non-spherical wave-fronts the measurement uncertainties can really be anything. Perhaps you can give an estimate for us?

Making plots or other (sanity) checks during an analysis is very important to make sure you don't have bugs or are not making mistakes. E.g. estimated

[2] Despite looping back, do not expect to see every sequence number in the data, because a device may send out on all channels, while we only measure on channel 6.

transmission powers shouldn't be bigger than 0 dBm unless the smart phone's owner hooked it up with an amplifier (which he didn't). These also help the reader 'believe' your results.

The system's x- and y-resolution you can estimate as the average of the uncertainties on the device's x- and y-position.

You decide what packets are closely enough spaced in time to be combined, but be sensible and be clear about your decision.