ECE 4999 Independent Project Final Report

Name: Hao Wu, NetID: hw622

Abstract:

Time Difference of Arrival (TDOA) is a technology which is mostly used to solve the locating problems for radar sensors and IOT networks. A locating problem is to compute the unknown position of a signal source which keeps sending signal to known position sensors. The TDOA technique, like its name, will use the time difference of signal's arrival at each sensor as the input data, to compute the potential position of the signal source. Usually, we can solve the locating problem with TDOA by geometrical way and numerical way. The geometrical solutions can be generated by computing the intersection points of group of hyperbola functions. Since only one solution is the correct position of the source, the strategy of choosing and aborting solutions may become hard and time consuming. For numerical methods, we would like to use the Newton method for quick searching the minimal point of our objective function, which is the correct position of the signal source. But the function may not be a convex function, meaning the choice of starting point of iteration counts a lot for the performance of our computing. This project probs into the distribution of starting points which can lead to the convergence to the correct location of the signal source and will use machine learning techniques to learn the pattern of this distribution.

Designs & Results:

1. Time Difference of Arrival

To clearly know what a locating problem is and how TDOA can help solve it, a figure is of the most help. The general and simplified setup of a locating problem is shown in figure 1.

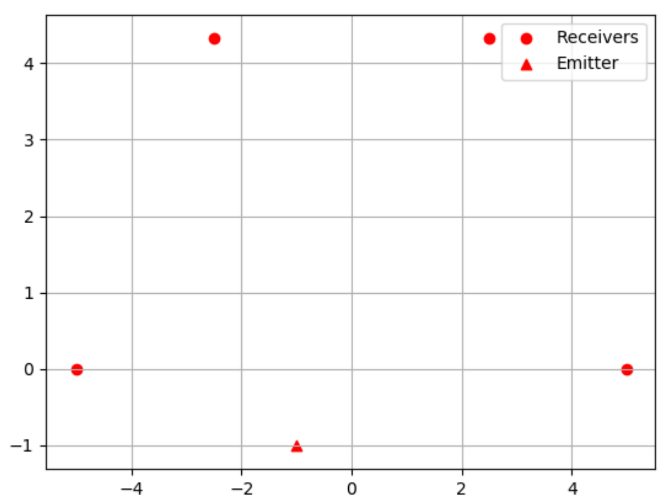

figure 1

We have four receivers, let say $R_1,\ R_2,\ R_3,\ R_4$, which receive signal from our emitter, let say E, and will keep record of the time the signal travels from E to each R, let say t_{ER_i}, i = {1, 2, 3, 4}. Let us suppose the signal travels at speed c, then we can have the following formulas:

$$
\begin{aligned}
\|E-R_1\| &= t_{ER_1}c \\
\|E-R_2\| &= t_{ER_2}c \\
\|E-R_3\| &= t_{ER_3}c \\
\|E-R_4\| &= t_{ER_4}c
\end{aligned}
$$

Since we are curious about the time difference, we shall select a reference receiver to measure the time differences of signal arrival from other receivers compared to the reference one. In here we can simply take the first receiver as the reference and obtain the TDOA formula:

$$
\begin{aligned}
\|E-R_2\|-\|E-R_1\| &= c(t_{ER_2}-t_{ER_1}) \\
\|E-R_3\|-\|E-R_1\| &= c(t_{ER_3}-t_{ER_1}) \\
\|E-R_4\|-\|E-R_1\| &= c(t_{ER_4}-t_{ER_1})
\end{aligned}
$$

For simplification, we can switch the subtraction of time arrival to a single notation:

$$
\begin{aligned}
t_{ER_2}-t_{ER_1} &= t_{R_2R_1} \\
t_{ER_3}-t_{ER_1} &= t_{R_3R_1} \\
t_{ER_4}-t_{ER_1} &= t_{R_4R_1}
\end{aligned}
$$

So now we will have

$$
\begin{aligned}
\|E-R_2\|-\|E-R_1\| &= ct_{R_2R_1} \\
\|E-R_3\|-\|E-R_1\| &= ct_{R_3R_1} \\
\|E-R_4\|-\|E-R_1\| &= ct_{R_4R_1}
\end{aligned}
$$

The function group above contains one solution of the real position of our emitter, so our mission under TDOA setup is to find that solution from the above function group.

2. Objective Function

We chose the objective function we want to minimize as the following:

$$
\min_E f(E) = \sum_{i=2}^{4}\left(\|E-R_i\|-\|E-R_1\|-ct_{R_iR_1}\right)^2
$$

in which case, we want to find an optimal E which can minimize the sum of squared error. The error is the caused by the wrong position of E, which can make the distance difference of E to R_i and E to R_1 different with the distance difference measured by time difference.

This objective function is not a convex function, which can be seen from the function plot in the following figures. A fact of this objective function is that when we get our iteration at the correct source position, the function value is zero, which is the minimal function value.

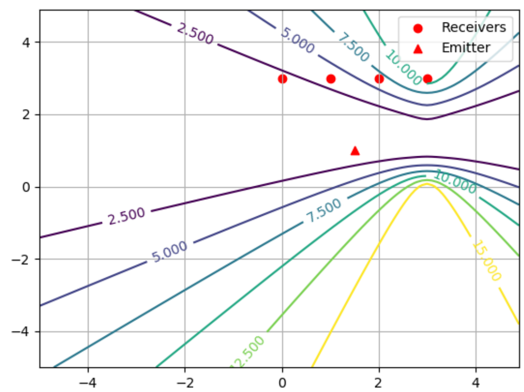

figure 2(a)

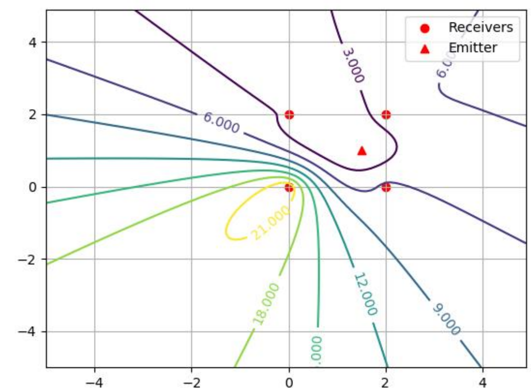

figure 2(b)

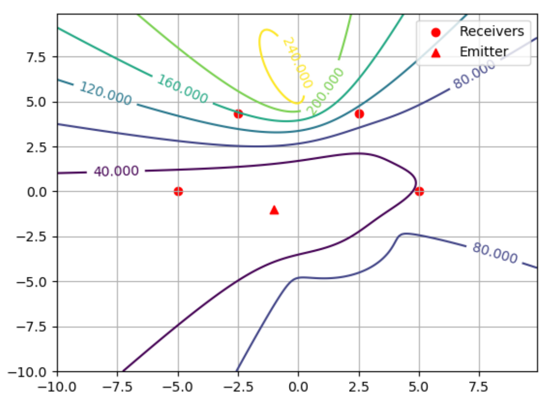

figure 2(c)

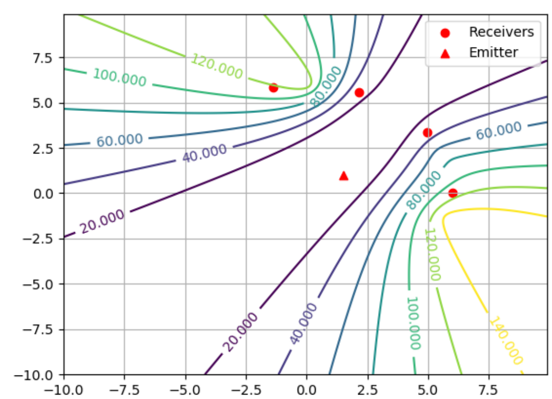

figure 2(d)

3. Newton Method for TDOA

The Newton Method is a common iterative numerical method for finding the minimizer of an objective function. Newton Method is a fast way to search a local minimum, but it will converge to other critical points when the Hessian matrix of objective function is not positive definite. Since our objective function is not a convex function, we cannot make sure that the Hessian is positive definite in the whole domain of this function.

We use the Newton method for our TDOA setup, by setting the iteration times limit equals to 200, the threshold of convergence equals to 1e-14. The starting points are chosen randomly in each 2x2 block of a 10x10 region. The results of a convergence map can be seen in the following figures:

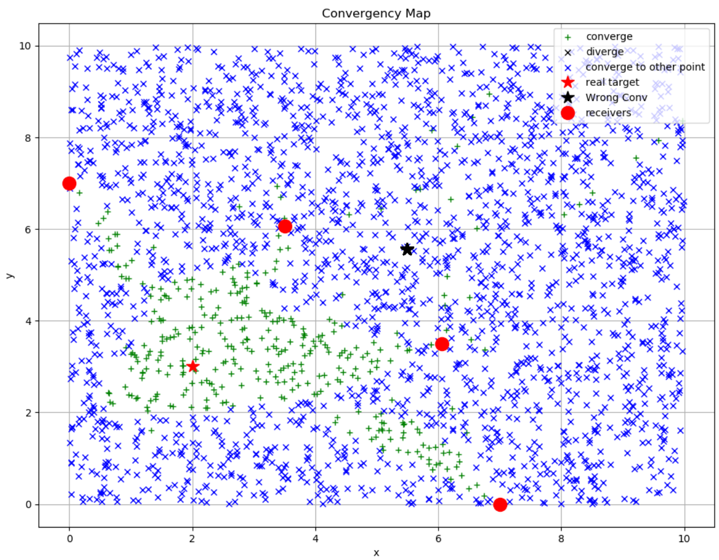

figure 3(a)

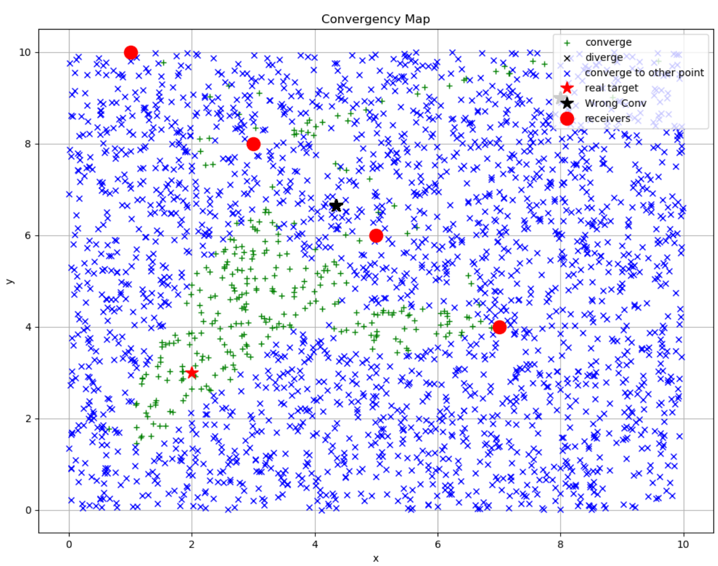

figure 3(b)

The green plus labels are those starting points which can converge to the correct source position, which is labeled as the red star. The blue cross labels are those starting points which cannot converge to the correct source position, but converge to other critical points, which is labeled as the black star. The red circles represent the receivers.

As we can see in figure 3(a) and figure 3(b), there is only a small region of green plus points, meaning the pure Newton Method works not that well to search the correct source position. The Hessian matrices of those blue cross starting points are not positive definite, leading the Newton Method to converge to the other critical points, the local maximum of our objective function. The reason that we are not satisfied with the result is in real world, let say four fire fighters want to locate the location of a victim which sending SOS signal, they only have small chances to get the convergence to real position of the victim.

So, we want to extend the region of green plus points, which means more chances of the searching will head to the real position of our target.

4. Modified Newton Method for TDOA

We use two techniques to modify the pure Newton Method. The first is Hessian modification and the second is trust region method. When we encounter the situation that Hessian matrix is not positive definite, we can do some simple modification on the Hessian to make the it a positive definite matrix. We first compute the eigenvalue of this Hessian matrix, then find which one is the smallest negative eigenvalue. Then, we can add a diagonal matrix with all the diagonal elements the same value which can help increasing the smallest negative eigenvalue to a small positive number.

For our situation, the Hessian is 2x2 matrix, which has only two eigenvalues λ_1 and λ_2. Then we compare which λ is a smaller one and having negative value. Since we want to compensate the negative value of eigenvalue, let say to make it from negative to 0.0001, then we need to add a diagonal matrix $(0.0001 + \lambda)I$ on the original Hessian. This method will make the Hessian become positive definite and somehow help to regulate the iteration to not go along for finding the maximum.

For trust region technique, we use a quadratic model function to approximate the original objective function in some range of a region, e.g. a circle area. We compute the gradient and modified Hessian at each iteration, and then compute a step size by solving trust region subproblem. If this step can generate a good approximation of the model function for our objective function, we accept the step and increase the radius of the trust region and move on. Otherwise we reject the step, and shrink our trust region.

The results of using these two method together are shown in the following figures:

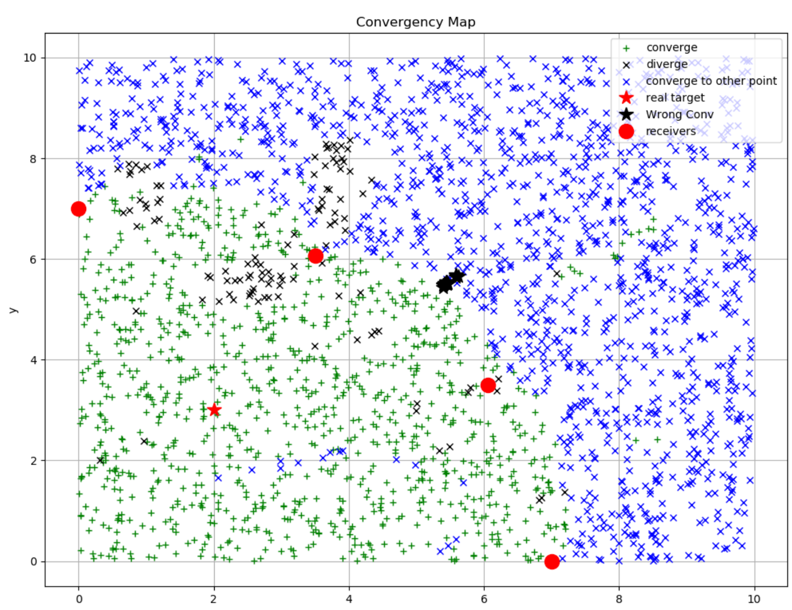

figure 4(a)

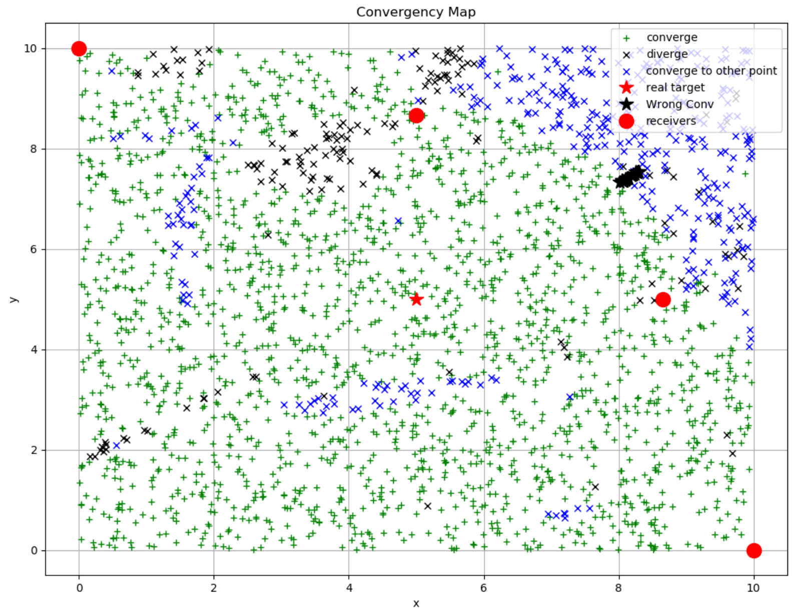

figure 4(b)

As we can see from figure 4(a) and figure 4(b), the number of green plus points increases and a much larger region of starting points can find the correct position of our target. However, we can also find some problems occur. There are some black cross points showing up in the figure, meaning that some starting points exhausted all iterations and cannot converge to any point. This happens when the quadratic model function always cannot make a good approximation for our objective function, and we keep on shrink our trust region.

Future Work:

The trust region method should work without any problem, meaning that the black cross points should not occur. After checking the rho parameter measuring the approximation of our model function, I find it always be negative for those points. A further study on this phenomenon is what I will do in the next step. Also, after finishing the part of generating correct dataset with trust region Newton Method, we can add in the machine learning part to analyze the pattern of the distribution of good starting points.