Optimization and Algorithms

Instituto Superior Técnico

Project 2023/24

1 Introduction

Knowing your location, or that of some asset of interest, is a key requirement that underlies many data collection and data processing applications. The near ubiquity of GPS often makes us take localization for granted, but in fact there are many practically relevant scenarios — indoor, underground, underwater, aerospace ... — where GPS is either unavailable or is not a technically viable option. In such cases, alternative (and typically simpler and less accurate) systems are used to provide location estimates. This project takes inspiration from systems for GPS-challenged environments and examines several issues from the perspective of optimization and algorithms.

The position of an asset (often referred to as a *target*, *source*, or *node*) in a 2D world under a suitable reference frame (i.e., a coordinate system) is denoted by $x \in \mathbb{R}^2$. In dynamic discrete-time scenarios over a finite horizon, the position at time $t \in \{1, 2, \dots T\}$ is denoted by $x(t)$.

Position estimates are derived from spatially-related measurements taken between the target and a set of reference points, or *anchors*, whose positions are known. The position of the k-th anchor is denoted by a_k (static case) or $a_k(t)$ (dynamic case). The *pairwise* measurements mentioned above may include ranges (i.e., distances), angles, Doppler shifts, etc. Additionally, a moving target may measure its own velocity relative to the reference frame independently from all other assets.

2 Source localization

Assume that a static target located at x^* determines its position from (noisy) measurements of its range to a set of M anchors, $r_k = \|x^* - a_k\| + \text{noise}$. A reasonable approach to determine a position estimate when measurements are inconsistent is to solve a (nonlinear) least-squares optimization problem that chooses x to approximately match measured ranges, r_k, and predicted ones, $\hat{r}_k = \|x - a_k\|$

$$\hat{x} = \arg\min_{x} \quad \sum_{k=1}^{M} (\hat{r}_k - r_k)^2 = \sum_{k=1}^{M} (\|x - a_k\| - r_k)^2 \tag{1}$$

When pertinent, we will denote the true (noiseless) range to an anchor as $r_k^* = \|x^* - a_k\|$.

Task 1. Consider a one-dimensional version of this problem with a single anchor ($M = 1$) located at $a = 2$ and range measurement $r = 1$. Plot the cost function $(|x - a| - r)^2$ and comment on its behavior.

Task 2. Repeat for the 2D case with two anchors located at $a_1 = (-1, 0)$, $a_2 = (3, 0)$ and range measurements $r_1 = 2$, $r_2 = 3$. Represent the cost function using contour lines. Is the minimizer unique? Ignoring "degenerate" configurations, how many anchors would you choose to get a cost function with a single global minimizer?

Problem (1) is not convex, so the powerful and generic convex optimization solvers that are currently widely available cannot be used to find the solution. This is not too bad in 2D but quickly becomes problematic in the higher-dimensional formulations of collaborative localization where multiple 2D locations are simultaneously sought. To harness the power of CVX and its solvers (http://cvxr.com/cvx/), we will use a convex approximation (i.e., a convex *relaxation*) of (1), namely,

$$\underset{x}{\text{minimize}} \quad \sum_{k=1}^{M} (\|x - a_k\| - r_k)_+^2 \tag{2}$$

the only difference being the use of $(u)_+ = \max(u,\ 0)$ before squaring.

Task 3. Repeat Tasks 1 and 2 for this relaxed cost function. Is the solution set more or less ambiguous than previously?

Task 4. Code the 2D problem of Task 3 in CVX (the function `square_pos` implements $(\cdot)_+^2$), solve it and comment on the results. Why didn't you have to supply an initial estimate of the solution?

2.1 Adding angle measurements

In addition to range measurements, several systems for GPS-less localization and navigation can provide pairwise angle measurements, i.e., information on the *direction* where one asset is located relative to a reference one. We assume that the direction is encoded as a unit-magnitude vector, expressed in a global coordinate system, that points in the relevant direction. We will incorporate angular information into the previously considered optimization problem by adding a penalization term to the cost function. This could be done in different ways, but we will use a simple method, suitable for moderate measurement noise: Given a reference point a and a measured angular direction u to the source, we will penalize the squared distance from a point x to the line with direction u that goes through a.

Task 5. Given a point x in arbitrary dimensions, quantify its distance to a line with direction u that goes through point a.

Consider the localization problem based on mixed range and angle measurements (r_k and u_k, respectively)

$$\underset{x}{\text{minimize}} \quad \sum_{k=1}^{M} (\|x - a_k\| - r_k)_+^2 + \lambda \|(I - u_k u_k^T)(x - a_k)\|^2 \tag{3}$$

where λ is a weight that balances the range and angle penalization terms. As for range measurements, we might denote as $\hat{u}_k$ the noiseless predicted direction for a source tentatively located at x, and as u_k^* the true direction for the source located at x^*.

Task 6. As you have seen, relaxing the range terms provides convexity but may increase the ambiguity of the solution of the localization problem. Adding angular information helps to disambiguate the problem. If all directions u_k are distinct, how many of them do you need so that the solution of (3) becomes unique regardless of the range terms? Why?

Task 7. In two twin plots represent the contour lines of the angular portion of the cost function for (i) a single anchor located at $a = (-1, 0)$ and u represented in polar coordinates by an angle of 40°; (ii) two anchors located at $a_1 = (-1, 0)$, $a_2 = (3, 0)$ as before and u_1, u_2 with angles 40° and 140°, respectively. Superimpose markers for the anchor positions.

Task 8. Modify your CVX implementation for Task 4 to include the two angular terms at 40° and 140° and solve it for $\lambda = 1$. How does the minimizer change from the one obtained in Task 4?

2.2 Adding motion

In dynamic scenarios all assets may move over time. Anchor positions $a_k(t)$ are known at the same time instants $t \in \{1, 2, \dots T\}$ as pairwise measurements, and the goal of localization is to recover the target trajectory $x(t)$. This could be tackled as a set of independent localization problems for different time instants, but that would not take into consideration relevant side information such as smoothness constraints on the trajectories. We assume that a target has onboard sensors which allow it to measure its own instantaneous velocity vector $v(t)$ with respect to the global reference frame. Then we could conceptually regularize the trajectory by adding to the cost function penalization terms of the form $\sum_t \|\dot{x}(t) - v(t)\|^2$. But since we work with discretizations for $t \in \{1, 2, \dots T\}$ of the true continuous-time trajectories, we need a way to approximate the unknown derivative $\dot{x}(t)$ for a given t in terms of the problem variables $x(1), \dots x(T)$. If measurements of the underlying continuous-time trajectory are sufficiently dense, velocity estimates at *discrete-time index* t are commonly obtained from backward, forward, or symmetrical differences

$$v(t) \approx \frac{x(t) - x(t-1)}{\Delta}, \qquad v(t) \approx \frac{x(t+1) - x(t)}{\Delta}, \qquad v(t) \approx \frac{x(t+1) - x(t-1)}{2\Delta}, \tag{4}$$

where it is assumed that the continuous-time trajectory is uniformly sampled, Δ (seconds) being the time difference between consecutive sampling instants. No other information is needed here on the mapping of discrete-time indices t to continuous time instants.

Task 9. Generate a simulated 2D trajectory for a target moving inside a "bounding box" such that each vertical or horizontal coordinate remains bounded between -20 and 20 m (this is a guideline, not a strict requirement). Combine lines and arcs to create "interesting" trajectories (lawnmower, racetrack, spiral, etc.). Place static anchors close to the vertices of the box. Then, simulate the target as it moves through the trajectory, e.g., at a constant speed of about 1 to 2 $\mathrm{m\,s^{-1}}$, and record/compute range, angle and velocity at a fixed rate of about 2 Hz. Collect a total of $T = 50$ to 100 samples for each type of measurement and make sure that your spatial discretization is not too coarse, otherwise your predictions of instantaneous velocity from consecutive positions of the target will be inaccurate.
Simulating the target motion may be accomplished in a number of ways, e.g., analytically for relatively simple trajectories. However, you may find it easier to use higher-level tools for generating trajectories such as those available in Matlab's Robotics System Toolbox.

In the reminder of this section we will focus on range-based localization, ignoring the angular measurements. However, it is important that you account for them in your simulation because they will be needed in Section 3. The following modification incorporates motion into the previously discussed localization problems. Note that the optimization variable x now concatenates the position estimates $x(t)$ for all $t \in \{1, 2, \dots T\}$

$$\underset{x}{\text{minimize}} \quad \sum_{t=1}^{T} \left(\sum_{k=1}^{M} (\|x(t) - a_k(t)\| - r_k(t))_+^2 \right) + \mu \|\hat{v}_t(x) - v(t)\|^2 \tag{5}$$

In (5) $\hat{v}_t(x)$ denotes one of the differences (4) to estimate the velocity at t from positions at present, past or future time indices. For $t = 2, \dots T-1$ it is usually best to use symmetric differences, but you should try and see what works best in your setup.

Task 10. Start by formulating the trajectory estimation problem in CVX, solve it for $\mu = 1$ and noiseless measurements, and check that your solution agrees with the actual trajectory. Then, multiply all velocity measurements by 0.8, which will induce an inconsistency; from the perspective of velocity measurements your trajectory will appear to be smaller than that inferred from range measurements. Solve the problem for $\mu \in \{0.01, 0.1, 1, 10, 100, 1000\}$, record the optimal trajectories, plot two of them to illustrate the impact of μ, and comment on the results.

Task 11. For the optimal solution obtained in Task 10 for each value of μ compute the range error and velocity error terms of the cost function

$$\mathrm{RE}(\mu) = \sum_{t=1}^{T} \sum_{k=1}^{M} (\|x(t) - a_k(t)\| - r_k(t))_+^2$$

$$\mathrm{VE}(\mu) = \sum_{t=1}^{T} \|\hat{v}_t(x) - v(t)\|^2$$

and create a 2D plot of RE vs. VE. This plot should look like a hyperbola. Explain its shape and prove that it is decreasing, i.e., if for two values μ_a, μ_b the optimal solutions satisfy $\mathrm{RE}(\mu_a) \leq \mathrm{RE}(\mu_b)$, then $\mathrm{VE}(\mu_a) \geq \mathrm{VE}(\mu_b)$.

Task 12. Retrieve your original (non-scaled) velocities for the simulated trajectory, and now assess the impact of noise on (5) for $\mu = 1$ and $\mu = 0$. For that, add Gaussian white noise to ranges and to each horizontal/vertical component of the velocities. Take 0.5 m as a reference standard deviation for range noise and $0.1/\sqrt{2}\,\mathrm{m\,s^{-1}}$ for each component of velocity noise. In your simulations you may scale all of these standard deviations by the *same* variable scaling factor to make the global scenarios more noisy or less so.
Quantify the error between the estimated trajectory, $\hat{x}$, and the real one, x, through the global mean navigation error

$$\mathrm{MNE}(\hat{x}) = \frac{1}{T}\sum_{t=1}^{T}\|\hat{x}(t) - x(t)\| \tag{6}$$

Is the error distributed uniformly across the trajectories? What happens as you reduce the number of anchors? You may optionally try to generate other trajectories to get a better sense of what geometries are easier or harder to track.

3 Localization with a single range

This section is motivated by the search and rescue efforts to find Malaysia Airlines flight MH370, which was lost on March 8, 2014 while traveling from Kuala Lumpur to Beijing. The plane veered off course shortly after takeoff and the onboard transponder, which usually broadcasts the plane's position to air traffic authorities throughput any flight, was switched off. After flying past the range of civilian and military radars in the area, all contact with the plane was lost except for a low-rate communication link with a single satellite, whereby telemetry information about the engines was transmitted to their manufacturer. The only available spatial information about the plane was extracted from those few telemetry packets, namely, the range to the satellite and the time derivative of that range, i.e., the *range rate*[1].

In this section we examine localization of a target from range and range rate measurements to a single anchor. Because this measurement model is much less informative than the ones considered in previous sections, we abandon the goal of estimating arbitrary trajectories, and instead focus on identifying the parameters for a target moving uniformly along a linear trajectory[2] with starting point x_0 and velocity v

$$x^*(t) = x_0 + vt \tag{7}$$

We wish to determine the linear trajectory from measurements taken at a set of time instants $\mathcal{T} = \{t_1, t_2, \ldots t_T\}$. Since the target is noncooperating, this has to be done without access to velocity or any other onboard inertial measurements. Note that, unlike in previous sections, here t is used directly in the motion equation (7), and is therefore more naturally expressed as a continuous-time variable (in seconds) rather than an integer index. With a minor abuse of notation, t will still be used below as an index into the set of measurements and into summations as $t \in \mathcal{T}$.

The anchor, located at a, collects noisy measurements over time of the ideal range and

[1] In practice, range rates were derived from the Doppler shift of communication packets received at the satellite. Range rates are also commonly used as one of the measurements in radar datection, where they are obtained from the Doppler shifts of high-energy pulses transmitted and reflected back from the target.

[2] The tenuous data available for flight MH370 suggests that its trajectory after flying past all land-based radar stations was also approximately linear, possibly crashing in a remote region of the Indian ocean west of Australia after running out of fuel.

range rate

$$r^*(t) = \|x^*(t) - a\| \qquad\qquad s^*(t) = \frac{dr^*(t)}{dt} \tag{8}$$

For a uniformly-moving target and static anchor, the range rate $s^*(t)$ is also commonly given by the inner product between the velocity vector v and the previously considered unit angular vector $u(t)$ that originates at the anchor and points toward the target. This is useful for generating synthetic range rate measurements from simulated trajectory data. The available observations are therefore $r_t = r^*(t) + \text{noise}$ and $s_t = s^*(t) + \text{noise}$. To jointly estimate x_0 and v we could formulate the nonlinear least-squares problem

$$\underset{x_0,\, v}{\text{minimize}} \quad \sum_{t \in \mathcal{T}} (\hat{r}(t) - r_t)^2 + \nu(\hat{s}(t) - s_t)^2 \tag{9}$$

where $\hat{r}(t) = \|x_0 + vt - a\|$ and $\hat{s}(t) = \frac{d}{dt}\|x_0 + vt - a\|$. As the range rate term $\hat{s}(t)$ is non-convex, problem (9) cannot be relaxed to a convex one by using $(\cdot)_+$ as before. Instead, we retain the non-convex cost function and resort to the Levenberg-Marquardt method to obtain a (local) minimum.

Task 13. Generate a simulated 2D linear trajectory for a target moving inside the same bounding box of Task 9 with velocity up to about $5\,\mathrm{m\,s^{-1}}$. Place the anchor sufficiently far from the target path to avoid numerical problems with singularity in the derivative $\hat{s}(t)$. Create $T = 50$ to 100 observations, adding (moderate) Gaussian white noise as before with standard deviation 0.5m for ranges and $0.1\mathrm{m\,s^{-1}}$ for range rates. For convenience, a file `measurements.mat` is provided with sample trajectory data and measurements.

Task 14. Represent the contour lines of the cost (9) as a function of the (unknown) 2D velocity vector, taking the initial position as the true x_0 and $\nu = 1$. Is the minimum located near the true value v? Are there other minima? If so, how do you interpret them? Optionally, also check (9) as a function of x_0 for true v (you don't have to report this).

Task 15. Explain why jointly estimating x_0 and v from r_t and s_t alone is intrinsically ambiguous. The ambiguity becomes less serious, but is not completely eliminated, if we fix one of the variables (as seen in the previous task). Below, we will take x_0 as the ground truth value and only search for the optimal v. This is akin to what was done for flight MH370, as investigators looking into its unknown flight path took the last available radar contact as the starting point.

Task 16. Implement the Levenberg-Marquardt method to solve the optimizaton problem (9) as a function of v with initial position fixed at the true x_0 and $\nu = 1$. What stopping criterion did you use? How does the outcome depend on the initialization of the algorithm? Plot the value of the cost function and the norm of the gradient across iterations.

4 Notes

4.1 Matlab implementation

For those who choose to implement the project in Matlab, vectorizing your code makes it both compact and efficient. Check out commands such as `pdist2`, `vecnorm`, `dot`, `repmat`,

`reshape`, `permute`, `squeeze`, `meshgrid`, `combinations` and various page-wise commands for matrices concatenated as n-dimensional arrays, `pagexxx`. Inline and anonymous functions are also useful, e.g., when used with automatic plotting commands such as `fplot`, `fcontour`. The `lsqnonlin` command for nonlinear least-squares minimization in Matlab's Optimization Toolbox includes the Levenberg-Marquardt algorithm and may be useful as a benchmark during code development and debugging.

4.2 Suggested schedule

Most working groups have 4 elements; share the load and work in parallel. For reference, here is a suggested timeline for 4 weeks to tackle the defined tasks. In each week, activities in different $\{\cdot\}$ could be handled in parallel:

1. $\{1, 2, 3, 4\}$ and $\{9\}$
2. $\{7, 8\}$ and $\{10\}$
3. $\{5, 6\}$, $\{11\}$, $\{12\}$ and $\{13\}$
4. $\{14, 15\}$ and $\{16\}$