Robust Orbit Determination

Bob Wilson

EE364b: Convex Optimization II Class Project

Introduction

Orbit determination is the process of estimating the motion of a satellite based on repeated measurements over time. Measurements may include the range—the distance from the satellite to a ground station—or the azimuth and elevation of the satellite. We focused on range measurements only. These measurements are nonlinear functions of the orbit; hence orbit determination is equivalent to inverting a system of nonlinear equations.

Let y_i be the ith range measurement (collected at time t_i), and let $f_r(x;t)$ be the distance from the ground station to a satellite with orbit x at time t. The goal of orbit determination is to find x so that $y_i \approx f_r(x;t_i)$ for all i.

Least squares

The traditional approach solves a sequence of least squares problems:

$$\text{minimize } \sum_{i=1}^{N}(y_i - \hat{f}_r(x;t_i))^2,$$

where $\hat{f}_r$ is the affine approximation to f_r centered at x^k. The solution, $x^\star$, forms the next linearization point x^{k+1}. This method is known as the Gauss-Newton algorithm and is in popular use in industry today.

Robust methods

Least squares methods are sensitive to non-Gaussian noise in the measurements. Real-life measurements often have fatter tails than Gaussian analysis would suggest—real data sets have outliers. We replaced the residual-squared penalty with a Huber penalty:

$$\phi_{\text{hub}}(r) = \begin{cases} r^2 & |r| \le M \\ M(2|r| - M) & |r| > M, \end{cases}$$

where M is a parameter defining the width of the quadratic region. This penalty is less sensitive to outliers than an ℓ_2 penalty.

Trust region penalty

Since we repeatedly linearize the (non-convex) f_r, solutions far from the linearization point are untrustworthy. We added a penalty on deviations from the linearization point, x^k. Putting it all together, we iteratively solved

$$\text{minimize } \sum_{i=1}^{N}\phi_{\text{hub}}(y_i - \hat{f}_r(x;t_i)) + \lambda\|W^{1/2}(x - x^k)\|_2^2,$$

where W is diagonal with entries corresponding to $\mathbf{diag}(J^TJ)$, with J the Jacobian. This idea was inspired by the Levenberg-Marquardt algorithm. We trade off between the competing objectives using λ, which is similar to a trust region size parameter. Compared to the least squares approach, this method is robust against outlying measurements and takes into account the fidelity of the affine approximation by discouraging solutions in directions where the function is changing quickly.

Trust region weight

We can adaptively choose the trust region weight, λ. If we anticipate the solution will have much smaller residuals, but in fact the residuals did not decrease by much, we likely deviated too far from the linearization point. If the predicted improvement was accurate, the linear approximation was good and we can try loosening the trust region penalty. Let

$$\hat{\delta} = \sum_{i=1}^{N}\phi_{\text{hub}}(y_i - f_r(x^k;t_i)) - \sum_{i=1}^{N}\phi_{\text{hub}}(y_i - \hat{f}_r(x^{k+1};t_i)),$$
$$\delta = \sum_{i=1}^{N}\phi_{\text{hub}}(y_i - f_r(x^k;t_i)) - \sum_{i=1}^{N}\phi_{\text{hub}}(y_i - f_r(x^{k+1};t_i)).$$

Here, $\hat{\delta}$ is the predicted improvement in the residual penalty, and δ is the actual improvement seen. If $\delta \ge \alpha\hat{\delta}$, for $0 < \alpha \le 1$, it means the true improvement in fit was comparable to the predicted improvement, validating our affine approximation. We decrease λ to $\lambda/\beta^{\text{succ}}$, where $\beta^{\text{succ}} > 1$, since we may be overly restricting the trust region. If $\delta < \alpha\hat{\delta}$, the true improvement did not live up to expectations, indicating our approximation is not very good. In this case, we increase λ to $\lambda/\beta^{\text{fail}}$, where $\beta^{\text{fail}} < 1$, to decrease the size of the trust region.

Huber penalty parameter

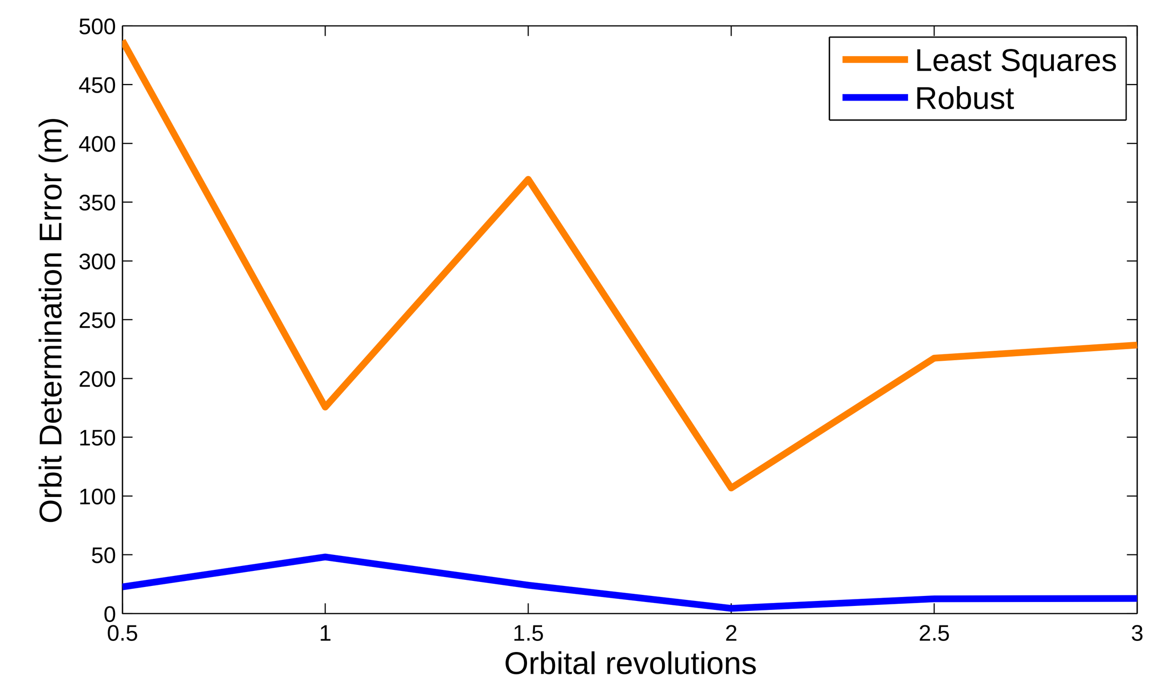
As we near convergence, the residuals mimic the noise distribution of the measurements. Hence, we can use the data itself to select the Huber penalty parameter, M. We modeled the data as a mixture of Gaussians. Most of the data is low-noise, with some large outliers. We used an Expectation-Maximization algorithm to estimate the noise for both groups, and chose M to be the standard deviation of the low-noise measurements. Unfortunately, this approach did not work well in practice, so we used a fixed M value of 0.015, corresponding to slightly larger than the data noise. There is room for future work here.

Other modeling details

We used a typical commercial injection orbit with a semimajor axis of $20,000$ kilometers. We used a simple two-body propagator both to generate the data and to solve the estimation problem. Atmospheric and light propagation effects were neglected. Ranging data from three ground stations in Los Angeles, Washington D.C., and Athens were simulated when the elevation of the satellite was greater than $10°$. Noise was added, with 90% of measurements having low noise (standard deviation 10 meters) and the remaining having high noise (standard deviation 1 kilometer). We scaled W so that the adaptively-chosen λ was between 1 and 100. This is equivalent to choosing the units of λ. In a real least squares problem, we would perform an orbit determination to identify the outlying data points, remove these, and re-optimize. This approach works extremely well in practice, but was not followed here. One of the benefits of our approach is that this second optimization is unnecessary.

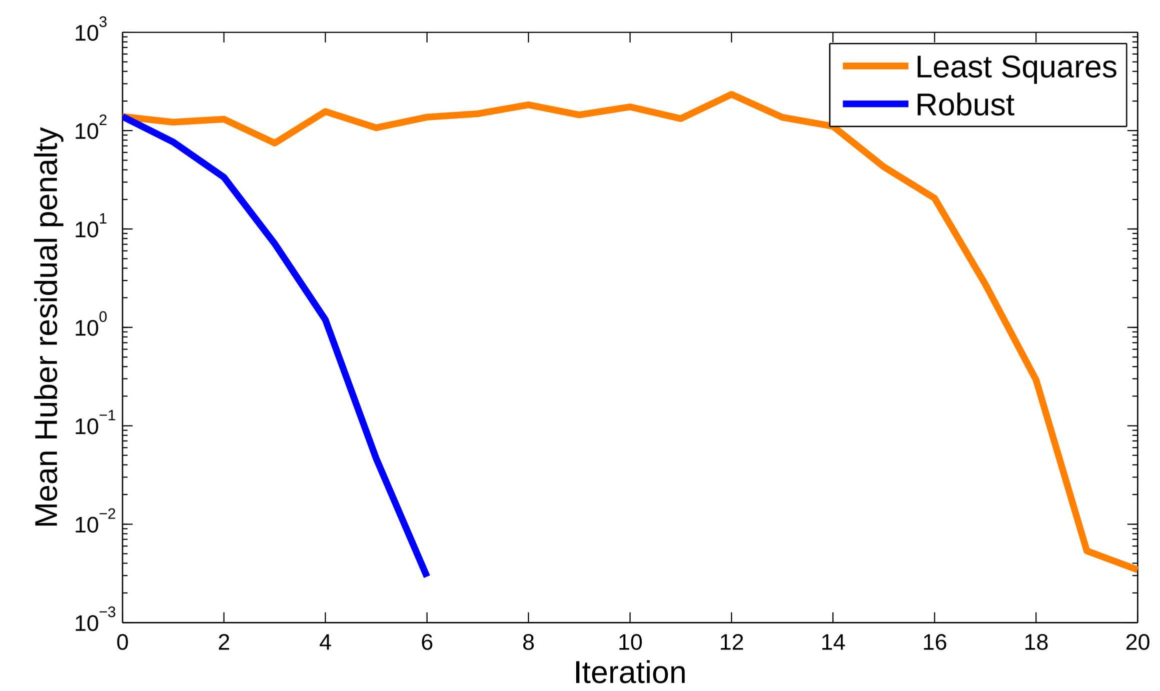
Accuracy

In practice, a full orbital revolution of data is needed before an accurate orbit determination may be performed using least squares methods. A method that gives an accurate answer more quickly than that is invaluable, since ranging data is expensive. We compared the performance of least squares and our robust method using amounts of data varying from 0.5 orbital revolutions to 3 revs (for a single instance for each data point). Even after half a rev of data, the robust penalty outperformed least squares using three full revs.



Convergence

In all cases examined, the robust problem converged more quickly than the least squares approach. This may be due to the resemblance of our trust region penalty to proximal methods, which have well-known convergence benefits. The figure shown uses three orbital revolutions of data.



Conclusion

With linear systems and many low-noise measurements, all methods work well. Sophisticated methods are most justified in nolinear or high noise situations, when relatively few measurements are available, or when fast convergence rates are valuable. In such situations, convex optimization techniques dramatically outperform least squares.