State Estimation - Assignment 5

Haniyeh Altafi
haniyeh.altafi@torontomu.ca

***Abstract*—This abstract presents an overview of various estimation methods and algorithms, including the Kalman Filter, Extended Kalman Filter (EKF), Particle Filter, RANSAC, Huber Loss, and Tukey's Biweight Loss. Each algorithm possesses unique characteristics and finds application in diverse domains. The abstract offers a concise depiction of their key differences, advantages, and disadvantages. These methods collectively provide a versatile toolkit for addressing challenges in state estimation, outlier handling, parameter estimation, and robust regression. The choice of method depends on factors such as linearity, noise distribution, the presence of outliers, and the level of robustness desired. This array of techniques equips practitioners to navigate complex real-world scenarios and derive meaningful insights across a spectrum of applications.**

I. Kalman Filter (KF)

The Kalman Filter is a mathematical tool used for estimating the state of dynamic systems based on measurements. It involves prediction and measurement update steps iteratively to refine state estimates. The KF excels in tracking linear systems and handling Gaussian noise efficiently. Nevertheless, it's limited by assumptions of linearity and Gaussian distributions, affecting performance in nonlinear or non-Gaussian scenarios. The KF algorithm comprises several steps:

1. Initialization: Define initial state, uncertainty, variables, and covariance matrix.

2. Prediction: Predict next state based on dynamics, updating uncertainty using covariance matrix.

3. Measurement Update: Combine predicted state with measurement using Kalman Gain for updated state and reduced uncertainty.

4. Repeat: Iterate steps 2 and 3 for each new measurement, continuously refining estimates.

II. Extended Kalman Filter (EKF)

The Extended Kalman Filter extends KF to nonlinear systems by linearizing via Jacobian matrix. Steps include initialization, prediction using nonlinear dynamics (linearized by Jacobian), and update with linearized measurement model. EKF handles nonlinearities but can struggle with high nonlinearity or uncertainty. Its steps are:

1. Initialization: Initialize state estimate, covariance matrices, and noise covariances.

2. Prediction: Predict state using nonlinear dynamics and Jacobian, update error covariance.

3. Update: Use measurements to refine estimate through linearized measurement model and Kalman gain.

4. Repeat: Iterate steps 2 and 3 for subsequent time steps.

III. Unscented Kalman Filter (UKF)

UKF overcomes KF's limitations in nonlinear dynamics and non-Gaussian noise. It samples deterministic sigma points for accurate nonlinear representation. UKF is great for nonlinear systems with non-Gaussian noise but requires more computation due to sigma point propagation. Steps:

1. Initialization: Initialize state, covariances, and noise.

2. Prediction: Augment state and covariances, generate sigma points, predict state using nonlinear process model.

3. Update: Transform sigma points through measurement model, update state using Kalman gain.

4. Repeat: Iterate steps 2 and 3 for each new measurement, refining estimate.

IV. ITERATED KALMAN FILTER

IKF handles uncertain measurement consistency using iteration. It refines state estimates via predicted and measured value differences, enhancing robustness against anomalies. IKF converges quicker with measurement noise but adds computational overhead. Quality initial estimates are crucial for its effectiveness. Steps:

1. Initialization: Set up initial state, covariance, and parameters.

2. Prediction: Predict state and update error covariance.

3. Update: Obtain measurements, refine estimate with residual error.

4. Repeat: Iteratively predict and update to refine estimate.

V. PARTICLE FILTER

The Particle Filter, known as Sequential Monte Carlo, estimates sequential systems using particles. It initializes particles, predicts their evolution, updates using measurements, and resamples based on weights. It's versatile for nonlinear, non-Gaussian cases but computationally intensive. Steps:

1. Initialization: Generate particles with equal weights.

2. Prediction: Move particles using system dynamics.

3. Update: Update particle weights based on measurements.

4. Resampling: Select particles based on weights, improving representation.

5. Repeat: Iteratively predict, update, and resample for changing states.

VI. HUBER LOSS

The Huber loss stands as a resilient loss function, adept at striking a balance between squared and absolute errors, making it a valuable tool in handling outliers. This loss function's primary application lies in scenarios where data may contain anomalies or noise that can significantly affect conventional loss functions.

The fundamental principle behind the Huber loss is to establish a smooth transition between quadratic behavior, akin to squared error loss, and linear behavior, reminiscent of absolute error loss. The "huber delta" parameter controls this transition, where residuals below the delta exhibit quadratic behavior, and those beyond it demonstrate linear characteristics.

The Huber loss offers several merits:

1. **Robustness**: The Huber loss showcases superior resistance to outliers compared to squared error loss, which can be disproportionately influenced by extreme values. 2. **Convergence**: In optimization algorithms, the Huber loss's intermediary behavior fosters smoother convergence, enhancing stability during the optimization process. 3. **Customization**: The huber delta empowers practitioners to tailor the loss's sensitivity to outliers, allowing for adaptable handling of diverse datasets.

However, some considerations arise:

1. Tuning: Selecting an appropriate huber delta value mandates a solid grasp of data distribution and outlier magnitude. The choice directly impacts the trade-off between robustness and sensitivity.

2. Optimal Delta: The optimal huber delta might vary across datasets. Experimentation and cross-validation are often necessary to pinpoint the most suitable value.

3. Sensitivity: While more robust than squared error, the Huber loss can still be sensitive to extreme outliers, especially when the huber delta is improperly set.

VII. TUKEY'S BIWEIGHT LOSS

Tukey's biweight loss is a potent tool in the realm of robust loss functions, designed to counteract the influence of outliers. This loss function systematically decreases the weight assigned to outliers, thereby mitigating their impact during parameter estimation.

Advantages of Tukey's loss function include:

1. Outlier Resilience: Tukey's loss function assigns minimal weight to extreme outliers, rendering it highly effective in minimizing the undue influence of such anomalies.

2. Saturation Mechanism: The loss function saturates after a designated threshold, which effectively curbs the sway of extreme outliers without entirely excluding them.

3. Reliability: Even in datasets with a significant proportion of outliers, Tukey's loss maintains the accuracy of parameter estimates.

However, some drawbacks should be considered:

1. Bias Possibility: The utilization of Tukey's loss function can introduce bias if the proportion of outliers is inaccurately estimated. This can affect the reliability of the resulting estimates. 2. Parameter Tuning: The parameter governing the saturation point is crucial for the method's efficacy. Careful fine-tuning is often necessary to achieve optimal performance. 3. Efficiency Trade-off: In situations where the majority of data points are not outliers, Tukey's loss may lead to suboptimal efficiency compared to alternative loss functions.

In conclusion, Tukey's biweight loss provides a robust means of handling outliers, offering a balance between the benefits of robustness and the preservation of data integrity.

VIII. RANSAC

The RANSAC (Random Sample Consensus) algorithm is a robust method for estimating the parameters of a mathematical model from a set of observed data points that may contain outliers. It is commonly used in image processing tasks, such as estimating geometric transformations like the homography matrix.

The RANSAC algorithm can be summarized as follows:

Random Sample Selection: A subset of data points is randomly chosen, typically consisting of corresponding points from the source and destination images.

Model Estimation: The selected points are used to compute the model parameters, involving solving linear equations, such as estimating a homography matrix.

Model Validation: The computed model is applied to transform the remaining data points from the source image to the destination image. The error between the transformed points and the actual points in the destination image is measured, often using the Euclidean distance.

Inlier Determination: Points with errors below a predefined threshold are considered inliers, indicating potential matches that fit the estimated transformation.

Iteration and Consensus: Steps 1 to 4 are repeated a sufficient number of times to identify the set of inliers that best fit the model. The set with the highest number of inliers is considered the consensus set.

After the iterations, the RANSAC algorithm returns the set of inliers that produced the highest number of inliers. The RANSAC algorithm robustly estimates the parameters of a mathematical model by repeatedly sampling and fitting the model, while mitigating the impact of outliers. However, it may struggle with datasets containing a high percentage of outliers or when multiple models need to be estimated simultaneously.

IX. Conclusion

In conclusion, these filtering techniques and robust loss functions offer a diverse array of tools for tackling a wide spectrum of challenges in state estimation, parameter estimation, and outlier handling across diverse domains.

TheKalman Filter (KF) forms the bedrock of dynamic system state estimation, excelling in linear systems and Gaussian noise scenarios. However, its effectiveness diminishes in the face of nonlinearity and non-Gaussian noise.

TheExtended Kalman Filter (EKF) extends KF's capabilities to nonlinear systems through Jacobian matrix linearization, though it grapples with approximation errors in highly nonlinear situations.

The Unscented Kalman Filter (UKF) sidesteps the limitations of KF and EKF with deterministic sigma point sampling, particularly well-suited for nonlinear dynamics and non-Gaussian noise. Yet, its increased computational demands and parameter tuning must be weighed.

The Iterated Kalman Filter (IKF) addresses measurement inconsistencies via iterative refinement, enhancing robustness against outliers, but at the cost of heightened computational complexity and sensitivity to initial estimates.

The Particle Filter provides a versatile approach for estimating state in sequential systems fraught with nonlinearities and non-Gaussian noise, despite its computational intensity.

Additionally, the Huber Loss furnishes a middle ground between squared and absolute error losses, showcasing robustness against outliers while facilitating smoother convergence compared to pure absolute error loss.

Tukey's Biweight Loss further bolsters robust regression, systematically diminishing outlier influence and demonstrating remarkable resilience. However, the need for parameter tuning and the potential introduction of bias warrants careful consideration.

Moreover, the RANSAC algorithm contributes a powerful methodology for estimating model parameters in the presence of outliers, particularly valuable in image processing tasks and scenarios involving geometric transformations. While robustly mitigating outlier impact, RANSAC might face challenges with high outlier percentages or simultaneous estimation of multiple models.

When it comes to estimation and robust regression, the appropriate choice among these techniques hinges on the unique attributes of each problem, encompassing factors like system linearity, noise distribution, and the prevalence of outliers. With this array of tools at our disposal, we are equipped to navigate the intricacies of real-world data and extract meaningful insights across a wide spectrum of applications.