LEARNING THE NUMBER OF PARTICLES IN NESTED FILTERING

SARA PEREZ VIEITES

The estimation of state-space dynamical systems with unknown parameters represents a significant challenge that has been approached by various methodologies. Among these solutions, probabilistic methods such as particle Markov chain Monte Carlo (PMCMC), sequential Monte Carlo square (SMC^2), and nested particle filter (NPF) stand out since they provide theoretical guarantees. However, nested filtering is the only method that is recursive, i.e., there is no need to reprocess the whole sequence of observations from scratch to update their estimates. This methodology is characterized by two intertwined layers of filtering, one for the parameter estimation and another one for the state estimation, aiming to estimate their joint posterior probability distribution. Although this methodology is better suited for long sequences of observations, the use of sampling techniques in both layers of filters (but especially in the parameter layer) still makes its computational cost prohibitive in high-dimensional problems.

In this talk, we will discuss several strategies to improve the efficiency to explore the parameter space and, thus, to improve the trade-off between computational cost and performance. One of the approaches is (1) to adapt online the number of parameter samples to significantly reduce computational cost without compromising performance. A new statistic is derived from the likelihood of the parameters to understand when samples are redundant or less informative, so they can be eliminated without reducing accuracy of the estimates. Another approach to this problem is (2) to relocate the parameter samples in those regions of the space that need further exploration to improve convergence for some given computational constraints (e.g., maximum number of samples). The samples are drawn based on the Bayesian Fisher information matrix (BFIM), that identifies which dimensions within the parameter space are more sensitive, i.e., have a higher influence on the model's output.