Fast Optimal Kalman Filter

Derive a new Kalman filter that gives optimal estimation accuracy subject to constraints on the memory and computational complexity of the algorithm. Standard Kalman filters give optimal accuracy for linear Gaussian problems, but they are oblivious to constraints on computational complexity and memory. In particular, we want the computer run time and memory to scale no worse than linearly with the dimension of the state vector for CPUs or GPUs. The computer run time for a standard textbook Kalman filter scales cubically with model size, which would be the kiss of death for deep learning or other such applications. Similarly, the memory for a standard Kalman filter scales quadratically with model size. We are interested in problems with extremely large dimensions of the state vector (e.g., millions or billions), which is common for training deep learning, weather forecasting, solutions of PDEs for fluid dynamics, electromagnetics, nuclear reactions and other industrial applications. Please make sure that your approximate covariance matrix of the state vector is positive definite, rather than merely positive semi-definite; this is crucial for stability of the Kalman filter.

Related algorithms for low rank Kalman filters are given in [1] to [6]. These papers approximate the covariance matrix as a diagonal matrix plus the product of low rank matrices, or approximate the information matrix this way, or approximate the matrix square root this way or approximate it using Kronecker factorization, etc. However, all of these methods are ad hoc, and none of these papers claim to give optimal accuracy for the resulting approximate algorithm subject to the constraint of linear scaling of memory and linear scaling of computational complexity with model size (i.e., the dimension of the state vector). We conjecture that a Kalman filter in Joseph form, which tells the math that we are using a sparse covariance matrix, might be optimal or nearly optimal.

Bonnabel [1] approximates the Kalman filter covariance matrix, whereas Chang [2] approximates its inverse using an SVD to pick the subspace corresponding to the biggest eigenvalues of the matrix. These are two very different measures of quality; Bonnabel throws away the subspace with the smallest eigenvalues, but Chang keeps this subspace and throws away the subspace with the biggest eigenvalues. The smallest eigenvalues correspond to the most accurately estimated states, and the biggest eigenvalues correspond to the states with the worst accuracy. Intuitively, for Kalman filtering it is a very bad idea to throw away the states with the best accuracy, and hence Chang's method should be better than Bonnabel's. This suggests that we would really like a hybrid of these two approaches, to get the best of both worlds. Hopefully, the new optimal algorithm that you derive will accomplish this.

To approximate covariance matrices it is tempting to minimize the Frobenius norm of the error, but this is not a good idea for Kalman filters. In particular, it does not guarantee that the approximate matrix is positive definite. Secondly, it is computationally extremely expensive. Thirdly, it does not exploit the structure of the Kalman filtering problem, as explained in [5].

Based on a quick reading, Pförtner's paper [8] appears to solve our problem completely, because it asserts that it gives an "optimal" solution. However, it throws away the subspace corresponding to the smallest eigenvalues of the error covariance matrix, just like Bonnabel does, and hence it is not actually optimal in any practical sense for Kalman filtering. Moreover, it does not use the Joseph form of the Kalman measurement update, and therefore it cannot be optimal in terms of accuracy of state vector estimation. Nevertheless, it is an interesting paper which has several useful ideas.

The memory and computer run time of Bonnabel's FA method scales linearly with model size, but unfortunately it also scales as the sixth power of the rank of the low-rank matrices [1]. This is not good enough for our applications. Please derive a new algorithm that scales no worse than Bonnabel's PPCA method in terms of memory and computer run time. In particular, Bonnabel's PPCA method scales linearly with model size and quadratically with rank of the low-rank matrices [1].

REFERENCES

[1] Silvère Bonnabel, Marc Lambert and Francis Bach, "Low-rank plus diagonal approximations for Riccati-like matrix differential equations," SIAM Journal 2024, https://arxiv.org/abs/2407.03373.

[2] Peter Chang, et al., "Low-Rank Extended Kalman Filtering for Online Learning of Neural Networks from Streaming Data," 2023, https://arxiv.org/abs/2305.19535.

[3] Jonathan Schmidt, et al., "The Rank-Reduced Kalman Filter: Approximate Dynamical-Low-Rank Filtering In High Dimensions," NIPS 2023, https://arxiv.org/abs/2306.07774.

[4] James Martens, et al., "Optimizing Neural Networks with Kronecker-factored Approximate Curvature," June 2020, https://arxiv.org/abs/1503.05671.

[5] Alessio Spantini, et al., "Optimal Low-Rank Approximations of Bayesian Linear Inverse Problems," July 2015, https://arxiv.org/abs/1407.346.

[6] Marvin Pförtner, et al., "Computation-Aware Kalman Filtering and Smoothing," March 2025, https://arxiv.org/pdf/2405.08971.