Automatic Differentiation for Tensor Based Target Tracking in Wide Area Surveillance Applications

Felix Govaers
Fraunhofer FKIE
Fraunhofer Str. 20, 53343 Wachtberg
GERMANY

felix.govaers@fkie.fraunhofer.de

ABSTRACT

Target tracking and Fusion Engines often are crucial components in modern military systems such as for instance for reconnaissance, surveillance, and electronic warfare. Despite the advances of machine learning approaches for various applications, for safety and security systems model based approaches are still preferred for the sake of explicability, interpretability, certifiability, and performance prediction in unforeseen scenarios. Since a closed- form of the Bayesian estimation approach is intractable in general, approximate methods have to be applied. Kalman or particle based approaches have the drawback of either a Gaussian approximation or a curse of dimensionality which both leads to a reduction in the performance in challenging scenarios. An approach to overcome this situation is state estimation using decomposed tensors. In this paper, the advantages of tensor decomposition methods based on automatic differentiation are demonstrated for target tracking in big data scenarios. Since the Canonical Polyadic Decomposition allows a highly effective and yet powerful representation of multi target information, the problem of state estimation can be applied even for a large number of targets in dense scenarios, in which conventional methods such as Gaussian Mixture based or particle based approaches would fail.

1. INTRODUCTION

In recent decades, sensor technology has become increasingly important for defense applications, and it is obvious that this trend will continue in the future. Defense-related sensors have conquered many novel applications, and existing applications have been brought to a much higher level of technical complexity. Nowadays, the distinction of information processing layers – once introduced to distinguish between sensor information and signal processing – was increasingly dissolved in favor of better performances. As a result, existing target tracking and wide-area surveillance methods do not address the challenges of today's sensor data in complex scenarios.

Due to the advances in nowadays sensor technology, the requirements and challenges for a data fusion engine have evolved since the very early years of this discipline in the 1960s [1]. Back then, data fusion referred to the compensation of the measurement error in air surveillance applications. This error indeed was uncorrelated white noise, since its dominant components were coming from thermal fluctuations. Moreover, the number of sensors involved was low – if not singular – and only a few objects were observed in parallel. In addition, the number of false alarms was high to achieve adequate detection probabilities. It can be seen that classical models for multiple target tracking are well suited to address these issues: Poisson distributed, independent and identically distributed clutter models and zero mean, Gaussian distributed noise allow analytical solutions of the underlying formulas and are therefore the basis for fast and reliable filtering models [2, 3].

However, challenges have changed. Today, often heterogenous sensors such as radar, lidar, E/O and I/R cameras and others are combined to achieve robustness and a high level of perception. Also, sensor data fusion is applied to a great variety of applications where for instance intelligence data from social media or

computer program logs for intrusion detection systems are combined to achieve situational awareness. Due to the high accuracy and resolution capacity of nowadays sensors, the number of observed objects is increased enormously, while the measurement error is often reduced to an almost irrelevant level [5, 6].

The key to robust and consistent target tracking in nowadays applications is the processing of large amounts of real-time sensor data using advanced, parallelizable sensor data fusion algorithms, which are optimized for big data processing [7]. Therefore, it is necessary to replace decades-old paradigms, which are based on the assumption that entities are most effectively represented in the form of parameterized probability densities. This is proven to be a promising strategy for a few to moderately many targets. In extreme scenarios with a high number of objects in a confined space, however, this is no longer feasible. For next generation methods of target tracking and extraction for big data applications, processing speed and parallelization must be given top priority, and simplifications and approximations must replace exact solutions whenever useful. In particular, it should be noted that the calculation of the extracted information relies on simple operations, allowing parallel use of GPUs.

In this paper, we present the application of automatic differentiation algorithms for tensorized data fusion probability densities. This provides the basis for information analysis and decision support so that target identification, classification or anomalies detection can be derived.

2. TARGET TRACKING IN BIG DATA

The theory of target tracking has exposed a growing family of algorithms to compute the *probability density function* (pdf) of a system state conditioned on noise corrupted sensor observations. An estimate of the state is then obtained for instance by taking the mean of the pdf. The corresponding covariance matrix which is the expected estimation error squared additionally provides a measure of accuracy for the estimate. Bayesian estimation is the framework of recursive filtering methodologies, which allows to process a current measurement data in terms of a prior or initial density and a measurement likelihood function, which statistically describes the performance of the sensor. Thus, a tracking algorithm is an iterative updating scheme for calculating a conditional a-posteriori pdf hat represents all available knowledge on the object state at the present time. The densities are explicitly conditioned on the sensor data time series. The iterative scheme consists of two processing steps per update cycle, referred to as prediction and filtering. The manipulation of the probability densities is given by the following basic equations (see [2, 3] for instance).

Prediction. Assuming the Markov property of the underlying target state, the prediction density is obtained by combining the evolution model with the previous filtering:

$$p(\mathbf{x}_{k-1}|\mathcal{Z}^{k-1}) \xrightarrow[\text{constraints}]{\text{evolution model}} p(\mathbf{x}_k|\mathcal{Z}^{k-1})$$

$$p(\mathbf{x}_k|\mathcal{Z}^{k-1}) = \int \mathrm{d}\mathbf{x}_{k-1} \quad \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k-1})}_{\text{evolution model}} \quad \underbrace{p(\mathbf{x}_{k-1}|\mathcal{Z}^{k-1})}_{\text{previous filtering}} .$$

Filtering. The a-posteriori pdf is computed by means of the Bayes theorem, which allows to combine the likelihood of the current sensor data with the prior density from the prediction step:

$$p(\mathbf{x}_k|\mathcal{Z}^{k-1}) \xrightarrow[\text{sensor model}]{\text{current sensor data}} p(\mathbf{x}_k|\mathcal{Z}^{k})$$

$$p(\mathbf{x}_k|\mathcal{Z}^{k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathcal{Z}^{k-1})}{\int \mathrm{d}\mathbf{x}_k \quad \underbrace{p(\mathbf{z}_k|\mathbf{x}_k)}_{\text{sensor model}} \quad \underbrace{p(\mathbf{x}_k|\mathcal{Z}^{k-1})}_{\text{prediction}}}.$$

According to this paradigm, an object track represents all relevant knowledge on a time varying object state of interest, including its history and measures that describe the quality of this knowledge. As a technical term, 'track' is therefore either a synonym for the collection of densities for all time instances within a trajectory or of suitably chosen parameters characterizing them, such as estimates and the corresponding estimation error covariance matrices.

The inherent assumption in the Kalman filter approach of a linear dependency between the measurement space of the sensor and the state space of the considered system is not given in most practical applications. As a consequence, numerical or analytical approximations have been proposed in the past decades in various forms. Most prominent might be the first-order Taylor approximation, the Extended Kalman Filter (EKF), which has low computational costs due to its analytic solution of the prediction and filtering steps (see [1] for instance). The performance of the linearization can be improved by means of deterministic samples chosen at the local neighborhood of the current estimate. This algorithm is known as the Unscented Kalman Filter (UKF). The term Particle filter (PF) subsumes all kinds of numerical solutions with non–deterministic samples. Here, knowledge about the state typically is represented by a set of state samples, which implies that the density is approximated by a Dirac– mixture. Because the process noise terms are simulated by means of appropriately sampled random vectors, these methods are also known as sequential Monte–Carlo (SMC) methods. However, but it soon became clear that this approach is hindered by the 'curse of dimensionality' in high dimensional problems. For this reason, PF approaches are usually not well suited for target tracking in big data since each track has to be represented with a high number of samples such that its density is approximated well enough. Existing multi target tracking filters are often based on Gaussian Mixtures (GM), since they allow an efficient computation of the multi target density or so-called "intensity" function. Actually these methods scale quite well in the case of well-separated targets, however, the numerical complexity grows exponentially in dense scenarios due to association problems or GM component explosions. As a consequence, these methods are not ready for nowadays technology, which provide sensor data with high resolutions and wide area observations.

3. TENSOR DECOMPOSOTION FOR TARGET TRACKING

In the recent past, tracking algorithms based on tensor decompositions have been proposed to overcome this problem [3-5]. Based on a discretization of the state space, the density becomes a multi-way tensor. The prior then is obtained by solving the Fokker–Planck–Equation (FPE). Challa and Bar-Shalom for instance use finite differences to obtain the solution of the FPE and show that a consistent result is obtained even for highly non–linear problems with large noise variances. This static approach has not become as popular as the particle filters due to the higher computational load. There is a notable change in the way of thinking since it was discovered that separated representations of discretized multi–dimensional functions have surprisingly good approximation properties. Nowadays it may even be seen as the only known way to overcome the curse of dimensionality. In other words, approximations by separable functions are of particular interest when the dimensionality of the problem becomes large. It is well–known that by means of a Canonical Polyadic Decomposition (CPD)[1] of a discretized density a computationally effective representation can be achieved [8].

[1] The CPD is also known as the CANDECOMP/PARAFAC decomposition.

The factorization of a N-way tensor into N factor matrices $\mathbf{A}^{(n)}$ with columns $\boldsymbol{a}^{(1)}{}_1, \ldots, \boldsymbol{a}^{(1)}{}_R$ is given by the sum of outer products of:

$$\mathcal{Y} \approx \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \ldots \circ \boldsymbol{a}_r^{(N)} = \hat{\mathcal{Y}}$$

This representation is schematically visualized in Figure 1 for $N = 2$ and $N = 3$, respectively.:

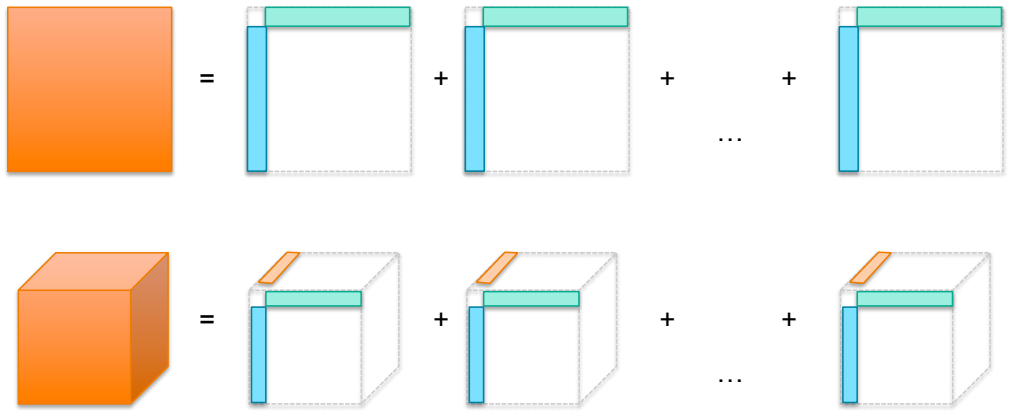

Figure 1: Schematic decomposition of a tensor by a factorization into a sum of outer products of vectors.

The time evolution of a density as a result of the Fokker-Planck Equation is shown in Figure 2. It can be seen that the density becomes broader while time passes, which reflects the stochastic nature of the underlying motion model.

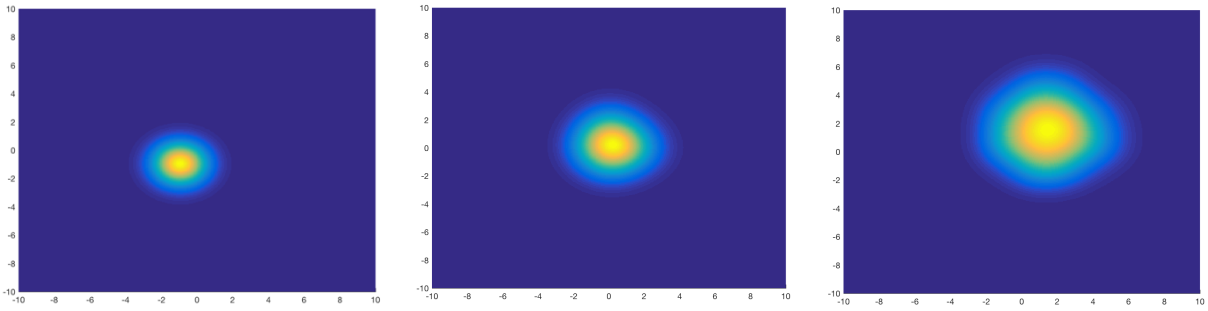

Figure 2: Exemplary time evolution of a target density in 2D using CPD tensors. The predicted tensor is computed as the solution of a Fokker-Planck Equation.

The filtering step, which includes the current measurements into the data fusion process, is obtained by a point-wise multiplication of the sensor model and the prediction density. The advantage of the CPD approach is that on the one hand the degree of approximation can well be adjusted to the performance of the underlying computation system: a smaller discretization step size yields a better representation of the information and more CPD components allow the computation of complex functions. On the other hand, the curse of dimensionality is avoided to a degree by means of the factorization along all dimensions. Also, complex associations of sensor data to track instances can be avoided by multi target point set methods. As a consequence, tensor decompositions are promising candidates for tracking targets in big data scenarios. An application in a multi target scenario is straight forward, since also intensity functions can be represented using tensor decompositions. These functions are the first-moment approximation of a multi target density,

that means they code the density function of the number of targets in the given field of view. The number of tracks can easily be obtained via integration, which can be done highly efficiently with tensor decomposition representations.

4. AUTOMATIC DIFFERENTIATION FOR FUSION ENGINES

The challenge of fusion engines is to incorporate heterogenous sensor information over time, where the measurements are informative with respect to a system state $\mathbf{x}_t$ of interest at a given time t. Often, dynamic systems such as objects with kinematic motion for instance are considered. A general model to describe the dynamic evolution of the system in time is the following stochastic differential equation:

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x})\mathrm{d}t + \mathbf{G}(\mathbf{x})\mathrm{d}\mathbf{w}_t,$$

where $\mathbf{f}$ is the drift vector, $\mathbf{G}$ is the diffusion coefficients matrix, and $\mathbf{w}_t$ is a Wiener process. For a given initial pdf of the state at time t_0, the time evolution is governed by the *Fokker-Planck Equation* (FPE) of the following form:

$$\begin{aligned}\partial_t p(\mathbf{x}) = &-\sum_{i=1}^{D} \partial_{x_i}(\mathbf{f}_i(\mathbf{x})p(\mathbf{x})) \\ &+ \frac{1}{2}\sum_{i,j=1}^{D} \partial_{x_i}\partial_{x_j}\left([\mathbf{G}(\mathbf{x})\mathbf{G}(\mathbf{x})^\top]_{i,j}p(\mathbf{x})\right).\end{aligned}$$

If the pdf p is given in CPD form and the FPE coefficients $\mathbf{f}$ and $\mathbf{G}$ can be represented in separable form, then one obtains an FPE operator such that

$$\frac{\partial p}{\partial t} = \mathbb{L}p$$

In the recent literature, two different approaches have been proposed to compute a numerical solution of the FPE for CPD tensors. The first solution is based on the numerical minimization, where the FPE operator is augmented with a differentiation matrix $\mathbf{D}_t$ for the time dimension, which is accumulated with the pdf [8, 9]:

$$(\mathbf{D}_t - \mathbb{L})p(\mathbf{x}, t) = 0.$$

However, the trivial solution $p = 0$ is included in the solution space. Therefore, additional constraint terms have to be added in order to have a non-trivial solution. Three such constraints have been identified in [7], one for initial value, boundary value and normality each. This leads to the new formulation of the minimization objective function:

$$\begin{aligned}\mathcal{R} = &||(\mathbf{D}_t - \mathbb{L})p(\mathbf{x}, t)||^2 + \alpha||\mathcal{N}p(\mathbf{x}, t) - p(\mathbf{x}, 0)||^2 \\ &+ \beta||\mathcal{M}p(\mathbf{x}, t)||^2 + \gamma||\mathcal{B}p(\mathbf{x}, t) - \underline{1}||^2\end{aligned}$$

Here, the first term is the FPE, the second enforces the correct boundary condition at $t = 0$, the third corrects the boundary conditions with respect to the state variables $\mathbf{x}$ and the last term is for the normalization such that p integrates to one at each time. In order to achieve a solution p in CPD form, the objective function has to be minimized with respect to the representation parameters $[\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]$.

The second approach uses the tensor exponential of the FPE operator [10], since it holds that

$$p(\mathbf{x}, t_k) = \exp\{\Delta_t \cdot \mathbb{L}\}\, p(\mathbf{x}, t_{k-1}),$$

where Δ_t is the time difference in the evolution step. Then, the tensor exponential is approximated by a Taylor series. During this process, the rank R of the CPD representation of p is increased. In order to have a CPD tensor of fixed rank, the result has to be approximated by a numerical decomposition technique again. Therefore, both approaches are heavily dependent on efficient decomposition techniques [11].

The sensor is assumed to produce measurements z at each time step according to the following non-linear model:

$$z = h(\mathbf{x}) + w,$$

where w is zero-mean Gaussian noise with standard deviation σ_z. If the sensor likelihood $p(z|\mathbf{x}_k)$ is evaluated on the discretized state space, it becomes a N–way tensor $\mathbf{L}$ with the same dimensions as the prior p. For the Bayesian update, it is required to have the likelihood in CPD form, too.

Written in tensors, the Bayes update is given by

$$\hat{\mathcal{P}} = \frac{\mathcal{P} * \mathcal{L}}{\sum_{i_1,\ldots,i_D} \mathcal{P} * \mathcal{L}\, \Delta_1 \cdots \Delta_D},$$

where * denotes the pointwise multiplication. Since the prediction as well as the update increase the rank of a CPD tensor, numerical methods have to be applied to decompose the tensor with a given rank R.

The working horse of tensor decomposition is the *Alternating Least Squares* (ALS) algorithm, which subsequently minimizes the cost function with respect to an alternating factor matrix $\mathbf{A}^{(n)}$, $n = 1, \ldots, N$. Due to the convexity of the cost function, this can be achieved analytically. To this end, the matricization $\mathbf{Y}_{(n)}$ of the tensor p with respect to mode n is computed:

$$\mathbf{Y}_{(n)} \approx \mathbf{A}^{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right)^T$$

Here, the dot symbol is the Khatri-Rao (column wise Kronecker) product. Taking the derivative with respect to $\mathbf{A}^{(n)}$ and setting it to zero yields the iterative update rule

$$\mathbf{A}^{(n)} \leftarrow \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) \left(\circledast_{k \neq n} \mathbf{A}^{(k)T}\, \mathbf{A}^{(k)} \right)^{\dagger}, \quad (n = 1, 2, \ldots, N),$$

where the following fact was used:

$$\left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right)^{\top} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) = \circledast_{k \neq n} \mathbf{A}^{(k)\top} \mathbf{A}^{(k)}$$

Here, the star symbol is the Hadamard (elementwise product), which is more efficient to compute. However, the ALS is known to be unstable for ill-conditioned problems. In data fusion applications, such conditions can easily appear, since often the density tensor contains many zeros for the state space where no probability

mass is present.

A more robust algorithm is the *damped Gauss-Newton* (dGN) [12], which can well be applied to find the minimum of the cost function. A straight-forward formulation can be achieved by vectorizing all parameters of interest:

$$a = \left(\mathrm{vec}\left[\mathbf{A}^{(1)}\right]^{\top} \ldots \mathrm{vec}\left[\mathbf{A}^{(N)}\right]^{\top} \right)$$

Analogously, the residuum vector **r** is defined as

$$\mathbf{r} = \mathrm{vec}\left[\mathcal{Y} - \hat{\mathcal{Y}}\right]$$

Let, in addition, **J** be the Jacobian with respect to the vectorized parameters. Here, automatic differentiation can be used to compute the gradient. Then, the iterative dGN step is given by

$$a \longleftarrow a + \left(\mathbf{J}^{\top}\mathbf{J} + \epsilon \underline{1}\right)^{-1} \mathbf{J}\mathbf{r}$$

where $\varepsilon > 0$ is a regularization parameter. After each update, the Matrix **J** and the vector **r** have to be computed.

5. CONCLUSION AND FUTURE WORK

In this paper, we have discussed the gradient based methods for tensor decompositions of multi variate probability densities in a discretized field of view. Tensor decompositions are a highly efficient method to compute and store the posterior information in a fusion engine centric application. The Bayesian recursion can be solved by means of the Fokker-Planck-Equation for the prediction and a point-wise multiplication with additional normalization for the update step. In each of step, usually the rank of a Canonical Polyadic Decomposition increases. This implies an increased volume of data to be processed and stored. Therefore, optimization based techniques have to be applied such that the rank is fixed at a given number, depending on the approximation quality trade off. Modern frameworks which include automatic differentiation can be used to automatically compute the gradients and to apply a gradient descent methodology.

BIBLIOGRAPHY

[1] F. Govaers, “Innovative Methods of Sensor Data and Information Processing in Big Data Scenarios,” in NATO Science and Technology Organization: Big Data and Artificial Intelligence for Military Decision Making, May 2018.

[2] Y. Bar-Shalom, X. Li, and T. Kirubarajan, Estimation with Applications to Tracking and Navigation. Wiley-Interscience, 2001.

[3] W. Koch, Tracking and Sensor Data Fusion: Methodological Framework and Selected Applications. Springer, 2014.

[4] B. Ristic, S. Arulampalam, and N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, 2004.

[5] S. Challa and Y. Bar-Shalom, "Nonlinear Filter Design using Fokker Planck Kolmogorov Probability Density Evolutions," IEEE Transactions on AeroSpace and Electronic Systems, vol. 36, no. 1, pp. 309–315, Jan 2000.

[6] G. Beylkin and M. J. Mohlenkamp, "Algorithms for Numerical Analysis in High Dimensions," SIAM J. Sci. Comput, vol. 26, pp. 2133–2159, 2005.

[7] B. N. Khoromskij, "Tensors-Structured Numerical Methods in Scientific Computing: Survey on Recent Advances," Chemometrics and Intelligent Laboratory Systems, vol. 110, no. 1, pp. 1 – 19, 2012.

[8] Y. Sun and M. Kumar, "Nonlinear Bayesian Filtering Based on Fokker Planck Equation and Tensor Decomposition," in 2015 18th International Conference on Information Fusion (Fusion), July 2015, pp. 1483–1488.

[9] M. A. Khan, M. Ulmke, B. Demissie, F. Govaers, and W. Koch, "Combining log-Homotopy Flow with Tensor Decomposition Based Solution for Fokker Planck Equation," in Proceedings of the 19th International Conference on Information Fusion, Istanbul, Turkey, July 2016.

[10] B. Demissie, F. Govaers, and M. A. Khan, "Non linear Filter Design using Fokker-Planck Propagator in Kronecker Tensor Format," in Proceedings of the 19th International Conference on Information Fusion, 2016.

[11] F. Govaers, "Time Evolution for Dynamic Probabilistic Tensors in Hierarchical Tucker Decomposition Form," in Proceedings of the IEEE Sensor Array and Multichannel Signal Processing (SAM), 2018.

[12] A. H. Phan, P. Tichavsky, and A. Cichocki, "On Fast Computation of Gradients for CANDECOMP/PARAFAC Algorithms," ArXiv e-prints, Apr. 2012.