



HAL
open science

Locally Optimal Confidence Ball for a Gaussian Mixture Random Variable

Pierre Sendorek, Maurice Charbit, Karim Abed-Meraim, Sébastien Legoll

► **To cite this version:**

Pierre Sendorek, Maurice Charbit, Karim Abed-Meraim, Sébastien Legoll. Locally Optimal Confidence Ball for a Gaussian Mixture Random Variable. 4th Int. Conf. on Indoor Positioning and Indoor Navigation(IPIN), Oct 2013, Belfort, France. hal-01002332

HAL Id: hal-01002332

<https://hal.science/hal-01002332>

Submitted on 13 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Locally Optimal Confidence Ball for a Gaussian Mixture Random Variable

Pierre Sendorek, Maurice Charbit *
 Karim Abed-Meraim**
 Sébastien Legoll***

* Télécom ParisTech, Paris, France
 pierre.sendorek@telecom-paristech.fr
 ** Polytech Orléans, Orléans, France
 *** Thales Avionics, Valence, France

Abstract—We address the problem of finding an estimator such as its associated confidence ball is the smallest possible in the case where the probability density function of the true position (or of the parameter to estimate) is a d -dimensional Gaussian mixture. As a solution, we propose a steepest descent algorithm which optimizes the position of the center of a ball such as its radius decreases at each step but still ensures that the ball centered on the optimized position contains the given probability. After convergence, the obtained solution is thus locally optimal. However our benchmarks suggest that the obtained solution is globally optimal.

Keywords — Confidence domain; Gaussian Mixture Model; Optimization; Monte-Carlo; Robust Estimation; Accuracy

I. INTRODUCTION

In navigation, it is often of practical interest to express the accuracy of a position estimator by the dimensions of its confidence domain [1]. One may ask which estimator achieves the optimal accuracy with respect to this criterion. In a Bayesian setting, when the probability density function (pdf) of the true position given the measurement is Gaussian, it is well known that the smallest ball containing the true position with a given probability is centered on the mean. In this case the best estimator is the mean. But the problem has less been studied when the probability density has less symmetries. However this situation naturally appears in navigation. When several sources are used to form the measurement vector, taking into account the probability of failure of each source results in obtaining a pdf of the position expressed as a Gaussian Mixture (GM) [2]. In this case it is interesting to have a position estimator such as its associated confidence ball is the smallest possible.

In this paper, we address the problem of finding the smallest confidence ball containing the position with a given probability. As a solution, we propose a steepest descent algorithm which optimizes the position of the center of a ball such as its radius decreases at each step but still ensures that the ball centered on the optimized position contains the given probability. After convergence, the obtained solution is thus locally optimal.

Our algorithm's solution is compared to the globally optimal solution (computed thanks to an exhaustive search) in the 1-dimensional case. It is shown that the globally optimal solution empirically matches our algorithm's. Thus, when the

probability density function is only a single Gaussian, the obtained solution matches the optimal solution and is the mean.

II. POSITION OF THE PROBLEM

A. Probability of being in a ball

Suppose that the pdf of our d -dimensional parameter of interest, say X , is described by a GM. Let N_g be the number of Gaussians composing the mixture, and for each component j from 1 to N_g , let v_j be the weight the Gaussian in the mixture, μ_j the mean of this Gaussian and C_j its covariance. The pdf of X thus writes

$$p_X(x) = \sum_{j=1}^{N_g} v_j \cdot f_j(x) \quad (1)$$

Where $f_j(x) = N(x; \mu_j, C_j)$ is the evaluation at x of the pdf of a Gaussian with a mean μ_j and with a covariance C_j . We call A the probability of X to be outside a ball of center c and of radius r . The definition of A is given by

$$A(c, r) = \Pr(X \notin B(c, r)) = \int_{x \notin B(c, r)} p_X(x) dx. \quad (2)$$

B. The Problem

The problem is to find a center c such as the radius r is the smallest under the constraint that X has to be in this ball with an expected probability of $1 - \alpha$. This problem is equivalent to the following :

Find c such as r is the smallest possible under the constraint $A(c, r) = \alpha$.

As the reader may have noticed, we chose to deal with the complementary of the probability to be inside a ball. This is because the floating representation of a number is more precise in the neighborhood of 0 than in the neighborhood of 1 and since α is usually closer to 0 than 1 in practical applications, it will be preferable, instead of using the cumulative function, to

use the complementary cumulative function which is more precise in this context.

III. THE OPTIMIZATION ALGORITHM

A. Overview

This section details the principle of our algorithm, which is the original contribution of this work. Derivation of the equations will be explained in further sections, which will be decreasing in the abstraction level.

Our algorithm proceeds to N_g steepest descents, each steepest descent being initialized with $c = \mu_j$. The initialization step requires a value of r such as $A(c, r) = \alpha$. This value of r can be found either by interval halving, or by more sophisticated methods such as the secant method or Newton method, because $r \mapsto A(c, r)$ is a decreasing function (as a complementary cumulative function). When both of these values are found, the optimization step starts by finding which of the (small) variations (ϵ_c, ϵ_r) of the couple (c, r) do not change the probability of X to be outside the ball. This gives a set of possible directions (ϵ_c, ϵ_r) such as

$$A(c + \epsilon_c, r + \epsilon_r) = \alpha. \quad (3)$$

Among all those possible directions for ϵ_c , we choose the one which leads to the greatest improvement in terms of radius : hence, the chosen direction is the *steepest*. The center c is optimized by being replaced by $c' = c + \epsilon_c$ and to finish the optimization step, instead of replacing r by $r + \epsilon_r$ as a radius for the following step, the algorithm solves the equality $A(c', r) = \alpha$ in the variable r (e.g. by one of the already mentioned methods) which is preferred to avoid the cumulation of linearization errors during the successive steps of the optimization. Once this optimization step is finished, another begins. The process is repeated as long as (the step is not negligible) there is an improvement of the radius.

B. Steepest descent direction

To find the steepest descent direction, we want to find which are the (small) variations (ϵ_c, ϵ_r) such as (3) is satisfied

which implies that we want

$$A(c + \epsilon_c, r + \epsilon_r) - A(c, r) = 0. \quad (4)$$

And because ϵ_r and ϵ_c are supposed to be small, we replace the left term by Taylor's first order approximation $A(c + \epsilon_c, r + \epsilon_r) = A(c, r) + \nabla_c A(c, r) \cdot \epsilon_c + \partial_r A(c, r) \cdot \epsilon_r$, where $\nabla_c A(c, r)$ is the gradient, which is the vector of partial derivatives according to the components of c and where

$\partial_r A(c, r)$ is the partial derivative of A according to r . Hence we get the equation

$$\nabla_c A(c, r) \cdot \epsilon_c + \partial_r A(c, r) \cdot \epsilon_r = 0 \quad (5)$$

or equivalently, since $\partial_r A(c, r)$ is negative

$$\epsilon_r = -\nabla_c A(c, r) \cdot \epsilon_c / \partial_r A(c, r) \quad (6)$$

Since several directions are possible, the problem is now to find the steepest descent direction of ϵ_c . To do this, we take among all the vectors ϵ_c which have the same (small) norm, say $|\epsilon_c| = \beta$, the one which minimizes ϵ_r . Finally, since Cauchy-Schwartz ensures the inequality

$$-|\nabla_c A(c, r)| \cdot \beta \leq \nabla_c A(c, r) \cdot \epsilon_c \leq |\nabla_c A(c, r)| \cdot \beta \quad (7)$$

As a consequence, the value of ϵ_r , which we want to be negative, is minimized when $\epsilon_c = -\beta \cdot \nabla_c A(c, r) / |\nabla_c A(c, r)|$ which saturates the left inequality in (7).

C. The algorithm

The value β is the size of the step which was supposed to be small during the calculations. But in practice, we take β so as to halve the dimensions of the actual radius and the algorithm works. Also, to avoid oscillations around local optima, when a variation of the center leads to an increase of radius (whereas the linearization "predicted" a decrease), the step is halved. Halving can be repeated at most N_h times, after which the algorithm considers that the potential improvement is negligible. This translates into the constraint $\epsilon_r = -r / Q$, which results in choosing

$$\beta = -\frac{r \cdot \partial_r A(c, r)}{Q |\nabla_c A(c, r)|} \text{ with an initial value } Q = 2. \text{ Finally,}$$

the algorithm to find the optimal (c, r) sums up to

```

 $r_{MC} = +\infty$ 
for(  $j \leftarrow 1 : N_g$  ){
     $Q = 2$ 
     $c \leftarrow \mu_j$ 
    find  $r$  such as  $A(c, r) = \alpha$ 
    do{
         $\beta \leftarrow -\frac{r \cdot \partial_r A(c, r)}{Q |\nabla_c A(c, r)|}$ 
         $\epsilon_c \leftarrow -\beta \cdot \nabla_c A(c, r) / |\nabla_c A(c, r)|$ 
         $(c_{old}, r_{old}) \leftarrow (c, r)$ 
    
```

```

    c ← c + ε
    find r such as A(c, r) = α
    if (rold < r)
        (c, r) ← (cold, rold)
        Q ← 2Q
    } while (Q ≤ 2Nh)
    if (r < rMC) {
        (cMC, rMC) ← (c, r) }
    
```

At the end of the algorithm, (c_{MC}, r_{MC}) will describe the locally optimal ball containing X with a probability $1 - \alpha$.

D. Computation of the probability to be in a ball

The computation of the value of $A(c, r)$ implies the use of the generalized chi-square cumulative function, which can be efficiently computed e.g. using the algorithms in [3,4]. Indeed

$$A(c, r) = \sum_j v_j \int_{x \in B(c, r)} f_j(x) dx \quad (8)$$

Where $\int_{x \in B(c, r)} f_j(x) dx = \Pr(\xi > r)$, for a variable ξ which follows a generalized non central chi-square law with appropriate parameters [4]. In the following sections, this value will be computed from its analytical formula for every presented algorithm.

E. Derivative according to the center

The expression of the gradient $\nabla_c A(c, r)$ has similarities with the expression of $A(c, r)$ which makes its Monte-Carlo computation comfortable. We derive the expression of the gradient by remarking that

$$A(c, r) = \int_{x' \in B(c, r)} p_X(x') dx' = \int_{x' \in B(0, r)} p_X(x' + c) dx' \quad (9)$$

Which allows to derive under the sign sum

$$\begin{aligned} \nabla_c A(c, r) \cdot \epsilon_c &= \int_{x' \in B(0, r)} \nabla_c p_X(x' + c) \cdot \epsilon_c dx' \\ &= - \sum_{j=1}^{N_g} \int_{x' \in B(c, r)} v_j \cdot f_j(x') \cdot (x' - \mu_j)^T C_j^{-1} \cdot \epsilon_c \cdot dx' \end{aligned}$$

F. Derivative according to the radius

The following derivations refer to a mapping of the Euclidean coordinates into the generalized d -dimensional polar coordinates. However, we won't need to explicit the integrals since the expression in polar coordinates will only be used to obtain the formula of the derivative according to the radius. The obtained integral has a pleasant expression to be mapped back into Euclidean coordinates, in which we will be

able to evaluate the integral numerically. Using (9) as a starting point, the substitution $\eta = x/|x|$ and $\rho = |x|$ leads to the generalized polar coordinates

$$A(c, r) = \iint_{\rho > 0} (\rho + r)^{d-1} p_X((\rho + r)\eta + c) d\rho \lambda(d\eta)$$

where λ is the Lebesgue measure on the unit sphere [5] which we won't need to make more explicit for calculating the derivative

$$\begin{aligned} \partial_r A(c, r) &= (d-1) \iint_{\rho > 0} \frac{(\rho + r)^{d-1}}{(\rho + r)} p_X((\rho + r)\eta + c) d\rho \lambda(d\eta) \\ &- \iint_{\rho > r} \sum_{j=1}^{N_g} \rho^{d-1} \left[\frac{\rho \eta^T}{\rho} C_j^{-1} (\rho \eta + c - \mu_j) \right] v_j f_j(\rho \eta + c) d\rho \lambda(d\eta) \\ &= \int_{x' \in B(c, r)} \sum_{j=1}^{N_g} \left[\frac{d-1}{|x' - c|} - \frac{(x' - c)^T}{|x' - c|} C_j^{-1} (x' - \mu_j) \right] v_j f_j(x') dx' \end{aligned}$$

IV. MONTE CARLO COMPUTATIONS

Our choice is oriented towards numerical integration since the analytical formulae of the derivatives are unknown to the authors in the general case. Among numerical methods, we chose Monte-Carlo which is known to be insensitive to the increase of dimensionality and is an efficient way to sample the integration space at points where the integrand has significant values (far from zero). Indeed, the derivatives $\nabla_c A(c, r)$ and $\partial_r A(c, r)$ can both be expressed, modulo the adequate choice of the functions g_j , as

$$\sum_{j=1}^{N_g} v_j \int_{x \in \mathbb{R}^d} f_j(x) \cdot g_j(x) dx$$

where each j^{th} term of the sum can be computed thanks to Monte-Carlo with Importance Sampling. Thus the integral

$$\int_{x \in \mathbb{R}^d} f_j(x) \cdot g_j(x) dx \quad (10)$$

is numerically computed by sampling the iid variables $(X_{j,t})_{t=1 \dots N_d}$ each one according to the law

$N(\mu_j, m C_j)$ where m is a value greater than 1. Such a choice of m favors realizations of $X_{j,t}$ outside $B(c, r)$

which is a set where the functions g_j are null. Thus the pdf of each random variable $X_{j,t}$ is $x \mapsto N(x; \mu_j, m C_j)$. Monte-Carlo approximates (10) by

$$\frac{\sum_{t=1}^{N_d} g_j(X_{j,t}) \frac{f_j(X_{j,t})}{N(X_{j,t}; \mu_j, mC_j)}}{N_d \cdot \frac{1}{N_d} \sum_{t=1}^{N_d} \frac{f_j(X_{j,t})}{N(X_{j,t}; \mu_j, mC_j)}}$$

which tends to the ratio of the expectations

$$\frac{\int g_j(x) \frac{f_j(x)}{N(x; \mu_j, mC_j)} N(x; \mu_j, mC_j) dx}{\int \frac{f_j(x)}{N(x; \mu_j, mC_j)} N(x; \mu_j, mC_j) dx'}$$

which is indeed the desired value (10), when N_d tends to infinity. However the strength of importance sampling in this case is that only a small amount N_d of drawings suffices to make the algorithm work, because the possible errors in the computation of ϵ_c and β at each step are approximately corrected at the next step thanks to the computation of the new values of ϵ_c and β which only take the current value of (c, r) in consideration as a starting point to the descent.

V. BENCHMARKS

In a mono dimensional setting, we compare the solution obtained by our algorithm to the globally optimal solution obtained by a greedy search on the discretized space. Our algorithm is assessed on 100 Gaussian Mixtures with randomly drawn parameters : N_g is drawn as $G + 2$ where G follows a geometric law of mean 4 (to avoid the trivial case $N_g = 1$), μ_j is drawn according to a centered Gaussian with a variance of 1 and the $C_j = K_j / 3$ (which are scalars in the 1D case) where K_j is drawn according to a chi-square law with 3 degrees of freedom. The variances are thus drawn of the same order of magnitude than the spacing between the means of the Gaussians. This is done so because the problem is easier when the variances are too small in comparison to the spacing between the means of the Gaussians. Finally the weights ν_j are drawn as

$$\nu_j = U_j / \sum_{i=1}^{N_g} U_i, \text{ where the variables } U_j \text{ are iid chi-squared}$$

variables with 1 degree of freedom. For each set of these parameters, we compare the obtained radius r_{MC} with the globally optimal radius r_G by computing their ratio. The evaluation is made on several sets of parameters (m, N_d, α) and for each one, a histogram of the ratio r_{MC} / r_G is made in figure (1). The radius obtained by our algorithm is in most of the cases the global optimum although our algorithm searches

for the best among N_g local optima. The use of importance sampling enables to converge to the right result with few (100) particles even when $\alpha = 10^{-7}$.

VI. CONCLUSION

This paper has proposed a position estimator under the form of an algorithm which minimizes its associated confidence ball in the case when the position's probability density function is expressed as a Gaussian mixture in multiple dimensions. The algorithm has been assessed in one dimension, where a comparison against greedy algorithm is possible. Numerical computations showed that the obtained confidence ball is the globally optimal one.

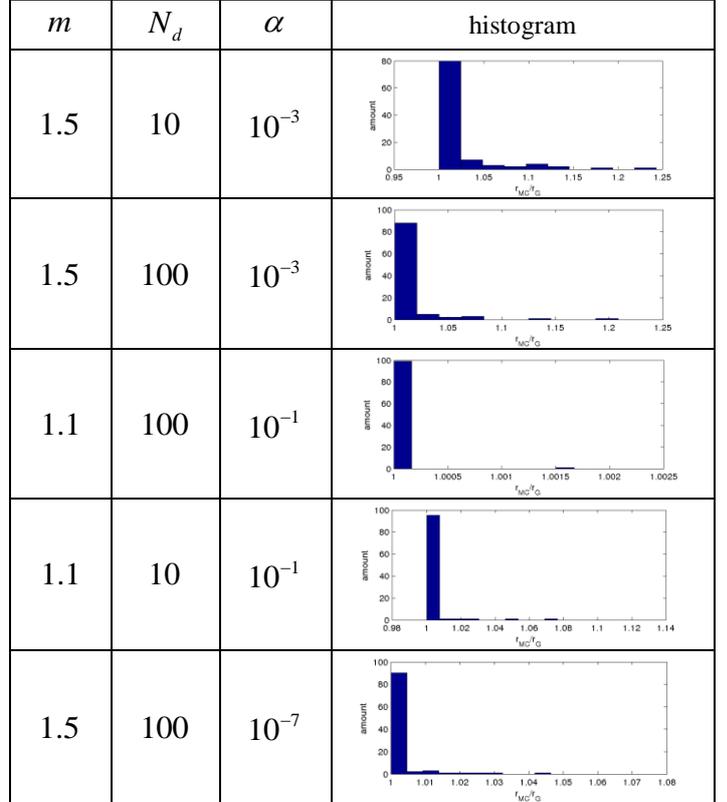


Figure 1. Histogram of the ratios of the obtained radius with the globally optimal radius.

REFERENCES

- [1] RTCA Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment. 1828 L Street, NW Suite 805, Washington, D.C. 20036 USA.
- [2] Pervan, Boris S., Pullen, Samuel P., Christie, Jock R., "A multiple hypothesis approach to satellite navigation integrity", *Navigation*, Vol. 45, No. 1, Spring 1998, pp. 61-84.
- [3] Robert B Davies "Numerical inversion of a characteristic function", *Biometrika* Trust, vol. 60, pp. 415-417, 1973.
- [4] Robert B Davies "Algorithm AS 155: The distribution of a linear combination of χ^2 random variables", *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, January 1980.
- [5] Wikipedia "Spherical Measure" http://en.wikipedia.org/wiki/Spherical_measure

