An Empirical Study into the Use of Chernoff Information for Robust, Distributed Fusion of Gaussian Mixture Models

Simon J. Julier*
ITT AES / Naval Research Laboratory
4555 Overlook Avenue SW
Washington, DC 20375, USA

Abstract — ***This paper considers the problem of developing algorithms for the distributed fusion of Gaussian Mixture Models through the use of Chernoff information. We derive a first order approximation and show that, in a distributed tracking problem in which sensor nodes are equipped with only range-only or bearing-only sensors, it yields consistent estimates.***

Keywords: Covariance Intersection, Chernoff Information, Gaussian Mixture Models.

1 Introduction

In many estimation problems, the assumption that random variables are independent, or that the correlations between them are known, is not true. Errors in the process and observation models, for example, lead to correlated process and observation noises [1]. Even if the models are known perfectly, computational and storage requirements often mean that the full correlation information cannot be maintained. One important class where this arises is in distributed data fusion (DDF).

DDF networks are composed of a set networked set of nodes. Nodes can fuse data acquired locally (from sensors) and remotely (from information propagated from other nodes). Because estimates rather than raw sensor data are propagated, the problem of double counting has to be avoided by factoring out common information [2]. When the network is known to lie in a tree-connected topology, a single path exists between any pair of nodes. This fact can be exploited to calculate the mutual information between the nodes using channel filters [3]. However, there are two important limitations with this approach. First, when the connection topology is arbitrary, channel filters cannot be used and, in fact, no local solution can be applied [4]. Second, when the estimates are not Gaussian, the factoring process used in a channel filter does not appear to have a closed form solution and computationally expensive numerical methods must be used instead [5]. An alternative approach, known as Covariance Intersection (CI), was proposed in [6, 7]. Given a set of estimates which are described by their means and covariances, CI provides a mechanism for fusing them together such that the estimate remains consistent. Although CI provides a very powerful and general method for fusing data in arbitrary networks, it only utilizes the mean and covariance of the estimates and cannot exploit any additional information about the probability distribution of the estimates.

Although the mean and covariance representation has been successfully used in many tracking systems, it has a number of limitations that can be encountered in a variety of practical contexts. For example, a single large mean and covariance is a poor representation of the uncertainty associated with a range-only sensor [8] or a bearing-only sensor [9]. Furthermore, multiple hypothesis tracking is poorly represented by a single large covariance. Therefore, we seek methods to generalize CI to exploit more information than a mean and covariance representation.

In this paper we develop an algorithm to extend CI to Gaussian Mixture Models (GMMs). Our algorithm is based on a first order approximation to the *Chernoff Information*. We describe our approach as empirical for two reasons. First, the only justification we have for using Chernoff Information as the basis of a generalization of CI are based on the observations by Mahler [10] and Hurley [11] (discussed in more detail below). Second, to develop a closed form solution, we use first order approximations of the Chernoff Information for GMMs and a simplified cost function for the optimization process. These approximations introduce their own sources of error and lead to extremely complicated error analysis. Rather than attempt to theoretically prove the properties of the algorithm, we demonstrate its performance on a distributed tracking application. Our results show that, despite these approximations, the algorithm is both consistent (in a mean squared error sense) and outperforms the only other algorithm we are aware of that tries to extend CI to GMMs [12].

The structure of this paper is as follows. Section 2 describes the distributed data fusion problem and the Chernoff Information solution. Our closed form approximation to Chernoff Information is developed in Section 3 and its properties are analysed. The perfor-

*The author is now with the Department of Computer Science, University College London. Email: S.Julier@cs.ucl.ac.uk

mance of the algorithm is illustrated in a distributed tracking problem in Section 4 and conclusions are drawn in Section 5.

2 Problem Statement

The problem of fusing data from two sources can be posed using Bayes Rule. Mathematically, this can be written as

$$P\left(\mathbf{x}_k|\mathbf{Z}^k\right) \propto \frac{P\left(\mathbf{x}_k|\mathbf{Z}_k^a\right) P\left(\mathbf{x}_k|\mathbf{Z}_k^b\right)}{P\left(\mathbf{x}_k|\mathbf{Z}_k^a \bigcap \mathbf{Z}_k^b\right)}$$

where $P\left(\mathbf{x}_k|\mathbf{Z}_k^a\right)$ is the probability distribution at node a, $P\left(\mathbf{x}_k|\mathbf{Z}_k^b\right)$ is that for b and $P\left(\mathbf{x}_k|\mathbf{Z}_k^a \bigcap \mathbf{Z}_k^b\right)$ is the common information between the two nodes. In an arbitrary network, where multiple paths can exist between a and b, $P\left(\mathbf{x}_k|\mathbf{Z}_k^a \bigcap \mathbf{Z}_k^b\right)$ cannot be calculated using local information, and the entire network must be considered. Given that the network can scale to thousands or millions of nodes, this erodes many of the advantages of distributed data fusion networks including their scalability, robustness and flexibility.

To overcome these limitations, a number of authors have attempted to develop methods that avoid the need to calculate the properties of the entire network. Mutamabara [13] and Berg [14], for example, developed methods in which only subsets of state information need to be distributed to subsets of nodes. Grime [3] developed algorithms for tree connected structures using channel filters considering mean and covariance representations. More generally, Chong and Mori [15] used graph theory to identify conditionally independent information within state estimates that is guaranteed to be independent and can be distributed amongst nodes. However, all of these solutions rely on specific assumptions about the network topology (it is tree-connected) and/or the structure of the state space (such that conditionally independent nodes can be identified). However, neither condition holds true for a general adhoc network with arbitrary, time varying system models.

In [6, 7], a data fusion algorithm called Covariance Intersection (CI) was presented. Suppose that the means and covariances of $P\left(\mathbf{x}_k|\mathbf{Z}_k^a\right)$ and $P\left(\mathbf{x}_k|\mathbf{Z}_k^b\right)$ are $(\mathbf{a}, \mathbf{A})$ and $(\mathbf{b}, \mathbf{B})$ respectively. Let the mean and covariance of the update be $(\mathbf{c}, \mathbf{C})$. Let $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ be the errors in the estimates. If the estimates are consistent in the sense

$$\begin{aligned} \mathbf{A} - \mathrm{E}\left[\tilde{\mathbf{a}}\tilde{\mathbf{a}}^T\right] &\geq \mathbf{0} \\ \mathbf{B} - \mathrm{E}\left[\tilde{\mathbf{b}}\tilde{\mathbf{b}}^T\right] &\geq \mathbf{0} \end{aligned} \tag{1}$$

where $\geq \mathbf{0}$ denotes positive semidefinite, the CI update rule is

$$\begin{aligned} \mathbf{C}^{-1} &= \omega\mathbf{A}^{-1} + (1-\omega)\mathbf{B}^{-1} \\ \mathbf{c} &= \mathbf{C}\left(\omega\mathbf{A}^{-1}\mathbf{a} + (1-\omega)\mathbf{B}^{-1}\mathbf{b}\right). \end{aligned}$$

This update is guaranteed to be consistent in the sense

$$\mathbf{C} - \mathrm{E}\left[\tilde{\mathbf{c}}\tilde{\mathbf{c}}^T\right] \geq \mathbf{0} \tag{2}$$

for $\omega \in [0, 1]$. The CI equations are equivalent to the Kalman filter equations with $\mathbf{A}$ replaced by $\mathbf{A}/\omega$ and $\mathbf{B}/(1-\omega)$.

Therefore, given consistent estimates, CI can yield consistent updates. However, the CI equations only use a linear update rule and can only utilize the first two moments of the state estimate. In many problems these representations are extremely crude and there is a strong incentive to consider how CI could be extended to utilize additional distribution information when it is available.

The first author to consider this issue was Mahler [10] who observed the following. Suppose $P_a(\mathbf{x})$ is a Gaussian distributed random with the pdf

$$P_a(\mathbf{x}) = \mathcal{N}\left\{\mathbf{x}; \mathbf{a}, \mathbf{A}\right\}.$$

Raising it to a power ω and renormalizing gives

$$P_a^{\omega}(\mathbf{x}) = \mathcal{N}\left\{\mathbf{x}; \mathbf{a}, \mathbf{A}/\omega\right\}.$$

In other words, the distribution is still a Gaussian with the same mean but the covariance has been scaled to $\mathbf{A}/\omega$. Similarly, calculating and renormalizing $P^{(1-\omega)}\left(\mathbf{x}_k|\mathbf{Z}_k^b\right)$ leads to a Gaussian-distributed random variable with mean $\mathbf{b}$ and covariance $\mathbf{B}/(1-\omega)$. Since CI resembles the KF with scaled covariance matrices, and since the KF is an application of Bayes' Rule with Gaussian distributed random variables, he extrapolated this observation to *all* distributions to give the expression

$$P_{\omega}(\mathbf{x}) = \frac{P_a^{\omega}(\mathbf{x})P_b^{1-\omega}(\mathbf{x})}{\int P_a^{\omega}(\mathbf{x})P_b^{1-\omega}(\mathbf{x})d\mathbf{x}}. \tag{3}$$

For $0 \leq \omega \leq 1$, one heuristic interpretation is that this tends to "flatten" the distribution. Because it becomes more uniform in nature, it becomes a more "conservative" estimate.

Mahler proposed choosing a value of ω to maximize the "peakiness" of the distribution,

$$(\omega, \hat{\mathbf{x}}) = \arg\sup_{\omega, \hat{\mathbf{x}}} P_{\omega}(\mathbf{x})$$

However, no actual studies were provided to show that this is a truly robust result and not simply the coincidence of the properties of Gaussian distributions.

Hurley independently made the same observations about Gaussians but noted that (3) is an equation used to calculate the *Chernoff Information* of a pair of distributions [11]. Chernoff Information quantifies the best achievable exponent in the Bayesian probability of error. It arises when constructing decision regions to minimize the probability of error and is extensively used in distributed target identification to determine the best achievable performance. Unlike Mahler's approach, the Chernoff Information is calculated across the entire distribution and is given by

$$C(P_1, P_2) = -\min_{0 \leq \omega \leq 1} \log\left(\int P_a^{\omega}(\mathbf{x})P_b^{1-\omega}(\mathbf{x})d\mathbf{x}\right) \tag{4}$$

The optimal value of ω, $\omega^\star$, has the property that

$$D^* = D\left(P_{\omega^\star}(\mathbf{x})||P_a(\mathbf{x})\right) = D\left(P_{\omega^\star}(\mathbf{x})||P_b(\mathbf{x})\right)$$

where $D(\cdot||\cdot)$ is the Kullback-Leibler divergence. In other words, the Chernoff Information is equally distance from both of the prior distributions[1].

Because Chernoff Information is a general result which applies to all probability distributions, it suggests that its relationship with CI is not a mere coincidence of the Gaussian form and thus might have general applicability. However, to our knowledge few authors have attempted to develop distributed data fusion algorithms using Chernoff Information. The only paper we are aware of which uses this approach is a study by Hwang [17] who compared several different approaches to distributed hypothesis testing for target identification. His study looked at the effects of sensors which misclassified data. His results suggested that Chernoff Information was worse in the sense that the estimate was more heavily affected by the incorrect sensors. However, these results can also be interpreted in a positive light — the Chernoff Information filter was more susceptible to the information because it did not converge as tightly to a single identification hypotheses.

These theoretical studies and empirical results suggest that Chernoff Information might offer a strategy for distributed data fusion. We now consider the application of Chernoff Information to an important class of probability distributions — the Gaussian Mixture Models.

3 Approximate Chernoff Information for Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a probability distribution function which can be written as the sum of a set of weighted Gaussian kernels. Therefore,

$$\begin{aligned} P_a(\mathbf{x}) &= \sum_{i=1}^{N_a} p_i \mathcal{N}\left\{\mathbf{x}; \mathbf{a}_i, \mathbf{A}_i\right\}, \\ P_b(\mathbf{x}) &= \sum_{i=1}^{N_b} q_i \mathcal{N}\left\{\mathbf{x}; \mathbf{b}_i, \mathbf{B}_i\right\}. \end{aligned} \tag{5}$$

We seek a closed form analytical approximation of the Chernoff solution such that

$$P_c(\mathbf{x}) = \sum_{i=1}^{N_c} r_i \mathcal{N}\left\{\mathbf{x}; \mathbf{c}_i, \mathbf{C}_i\right\}. \tag{6}$$

[1]This is distinct from O'Brien's Fusion of Correlated Probabilities (FCP) algorithm [16]. This algorithm used the expression

$$P_{\alpha,\beta}(\mathbf{x}) = \frac{P_a^\alpha(\mathbf{x}) P_b^\beta(\mathbf{x})}{\int P_a^\alpha(\mathbf{x}) P_b^\beta(\mathbf{x}) d\mathbf{x}}.$$

No conditions were placed on the values of α and β and thus it could be interpreted as a generalization of Chernoff Information. However, no theoretical analysis has been provided for this form.

There are three reasons why we consider pdfs of this form. First, GMMs are a very natural extension of the mean and covariance representation. Second, the family of GMMs is, in principle, extremely general and almost any pdf can be expressed precisely using GMMs. Furthermore, many distributions can be approximated well by a small number of terms of a GMM. Therefore, it is an extremely important practical distribution. Finally, GMMs share a strong theoretical relationship with Multiple Hypothesis Tracking (MHT). As we discuss in the conclusions, MHT might offer a mechanism for generalizing these results even when the mixtures are not Gaussian distributed.

Upcroft proposed a form of CI for GMMs which we term the Pairwise Component CI (PCCI) fusion rule [12]. Given the two distributions $P_a(\mathbf{x})$ and $P_b(\mathbf{x})$, CI is applied to each pair of estimates in turn. Let ω_{ij} be the weight applied to the fused estimate from the ith component of $P_a(\mathbf{x})$ and the jth component of $P_b(\mathbf{x})$. Then $N_c = N_a N_b$ and the ijth component is given by

$$\begin{aligned} \mathbf{C}_{ij}^{-1} &= \omega_{ij}\mathbf{A}_i^{-1} + (1-\omega_{ij})\,\mathbf{B}_j^{-1} \\ \mathbf{c}_{ij} &= \mathbf{C}_{ij}\left(\omega_{ij}\mathbf{A}_i^{-1}\mathbf{a}_i + (1-\omega_{ij})\,\mathbf{B}_j^{-1}\mathbf{b}_j\right) \\ r_{ij} &= \frac{\omega_{ij}p_i + (1-\omega_{ij})\,q_j}{\sum_{k=1}^{N_a}\sum_{l=1}^{N_b}\omega_{kl}p_k + (1-\omega_{kl})\,q_l} \end{aligned} \tag{7}$$

Heuristically, this form is motivated by the fact that when $\omega_{ij} = 1$ the estimate should only contain the component from A whereas if $\omega_{ij} = 0.0$ then the estimate should only contain a component from B. Furthermore, CI can be applied to each component independently. However, this form is an extremely poor approximation to the Chernoff solution. This is illustrated in Figure 1. The figure shows contour lines of the pdfs of two input estimates, the Chernoff Information solution (calculated numerically) and the PCCI when ω_{ij} is chosen to minimize the determinant of $\mathbf{C}_{ij}$. As can be seen, the Chernoff Information solution has a single strong mode whereas the PCCI maintains multiple modes. Furthermore, the PCCI tends to underweight the middle mode which is closest to the Chernoff solution.

This can be quantified using a metric proposed by Comaniciu [18]. The metric quantifies the distance between two distributions and is given by

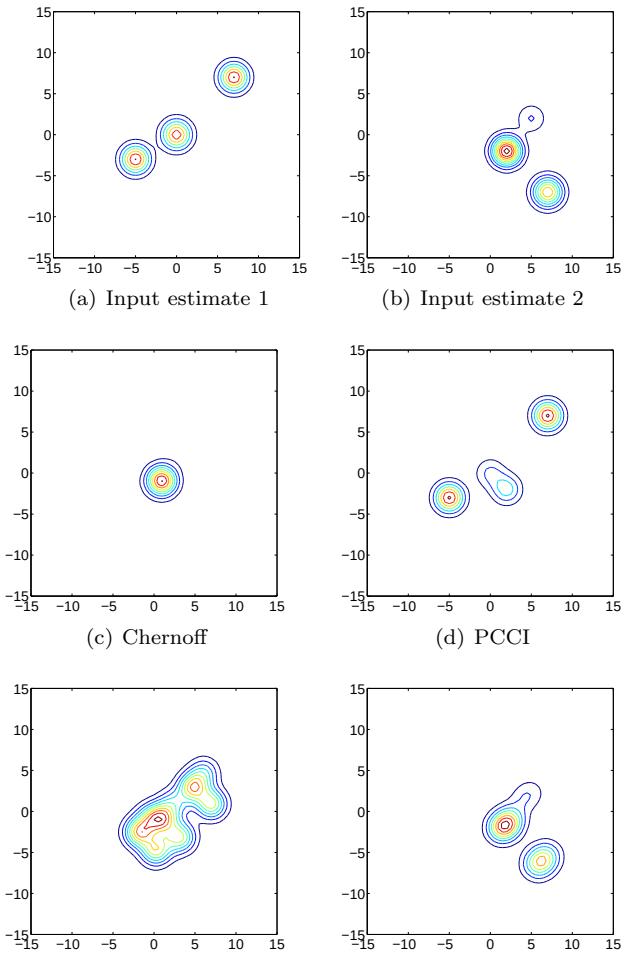
$$d = \sqrt{1 - \rho\left[P(\mathbf{x}), \hat{P}(\mathbf{x})\right]}$$

where

$$\rho\left[P(\mathbf{x}), \hat{P}(\mathbf{x})\right] = \int \sqrt{P(\mathbf{x})\hat{P}(\mathbf{x})}dx$$

is the Bhattacharyya Coefficient. This metric has the property that its value is lies between 0 and 1. The results for this example are shown in Table 1. The metric has a high value of 0.73.

Given this deficiency, we seek a more accurate calculation of the Chernoff Information. However, (3) does not have a closed form solution for a GMM. Its value could be approximated numercally using a grid.



(a) Input estimate 1 (b) Input estimate 2 (c) Chernoff (d) PCCI (e) Pseudo-Chernoff 1 (f) Pseudo-Chernoff 2

Figure 1: Contour plots of pdfs for different fusion algorithms.

Algorithm	Cost
PCCI	0.7286
Pseudo-Chernoff 1	0.6608
Pseudo-Chernoff 2	0.6347

Table 1: The costs of the different approximations.

Although the sparse methods proposed by Bucy and Senne can be applied [19] to reduce the computational costs, the curse of dimensionality means that, in general, this approach is prohibitively expensive. A similar difficulty arises in evaluating (4).

In this paper we use two approximations. First, to calculate the power series we use the approximation

$$\left(\sum_{i=1}^{n} d_i\right)^{\omega} \approx \sum_{i=1}^{n} d_i^{\omega}.$$

Substituting into (5),

$$P_a^{\omega}(\mathbf{x}) = \frac{1}{\sum_{i=1}^{N} p_i^{\omega}} \sum_{i=1}^{N} p_i^{\omega} \mathcal{N}\left\{\mathbf{x}; \mathbf{a}_i, \mathbf{A}_i/\omega\right\}. \tag{8}$$

In other words, it leads to an N component GMM. The means of each component remain the same but all the covariances are scaled up by a factor of $1/\omega$ and all weights have been renormalized. A similar expression is used to calculate $P_b^{(1-\omega)}(\mathbf{x})$.

Therefore, the update rule has $N_c = N_a N_b$ components and can be written as

$$\begin{aligned} \mathbf{C}_{ij}^{-1} &= \omega \mathbf{A}_i^{-1} + (1-\omega)\,\mathbf{B}_j^{-1} \\ \mathbf{c}_{ij} &= \mathbf{C}_{ij}\left(\omega \mathbf{A}_i^{-1}\mathbf{a}_i + (1-\omega)\,\mathbf{B}_j^{-1}\mathbf{b}_j\right) \\ r_{ij} &= \frac{p_i^{\omega} q_j^{(1-\omega)}}{\sum_{k=1}^{N_a}\sum_{l=1}^{N_b} p_k^{\omega} q_l^{(1-\omega)}} \end{aligned} \tag{9}$$

The effect of this approximation is illustrated in Figure 1(e). The results were calculated using the first order expansion approximation and the value of ω calculated by the numerical Chernoff solution. As can be seen, the estimate consists of a large, interconnected mass whose main peak lies in the same location as the Chernoff solution. Despite the fact that the solution does not have any well-defined modes, Table 1 its coefficient is smaller than that for PCCI.

The second difficulty is to approximate the calculation of ω such that (4) is satisfied. This could be calculated by Monte Carlo integration [20] or by adapting the distance approximation developed by Goldberger to measure the dissimilarity between two GMMs [21][2]. However, this introduces its own approximations and we do not investigate their effects here. Rather, we use the (somewhat crude) approach of minimizing the covariance of the entire mixture of the distribution,

$$\begin{aligned} \mathbf{c} &= \sum_{i=1}^{N_a}\sum_{j=1}^{N_b} r_{ij}\mathbf{c}_{ij} \\ \mathbf{C} &= \sum_{i=1}^{N_a}\sum_{j=1}^{N_b} r_{ij}\left(\mathbf{C}_{ij} + \mathbf{c}_{ij}\mathbf{c}_{ij}^T\right) - \mathbf{c}\mathbf{c}^T. \end{aligned} \tag{10}$$

The results of the Pseudo-Chernoff algorithm is shown in Figure 1(f). This distribution possesses two distinct modes, one over the Chernoff solution, the other offset to the right. The cost metric shows that the approximation has a smaller cost than using ω calculated by Chernoff and PCCI, and thus shows that it is a more accurate approximation.

We now show the effect of this suboptimal solution in a target tracking example.

4 Example

A sensor network, consisting of the five nodes listed in Table 2, attempts to estimate the position and velocity of a target in 2D. Each sensor has its own detection range and sensor error characteristics. Nodes 1, 2 and

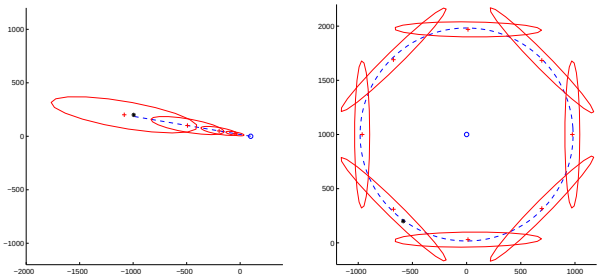
[2] Both of these methods depend on the observation that

$$C = \int P_a^{\omega}(\mathbf{x}) P_b^{(1-\omega)}(\mathbf{x}) d\mathbf{x} = \int P_a(\mathbf{x}) \left(\frac{P_b(\mathbf{x})}{P_a(\mathbf{x})}\right)^{1-\omega} d\mathbf{x}$$

In other words, the expectation can be taken with respect to $P_a(\mathbf{x})$. Since this probability distribution is known it does not have to be approximated.

Node	Position	Velocity	Type	Range	Uncertainty
1	(0,0)	(0,0)	Bearing	2000	0.5°
2	(100,0)	(-1,0)	Bearing	2000	2°
3	(0,1000)	(0,-2)	Bearing	200	1°
4	(0,-1000)	(0,0)	Range	800	10
5	(0,1000)	(0,0)	Range	1200	10

Table 2: The location, velocity, type detection range and accuracy for the sensors used in the example.



(a) Bearing-only sensor. (b) Range-only estimate.

Figure 2: The GMMs generated by the different sensor types. The location of the target is given by $\star$ and the location of the node by $\circ$. Sensor measurements are the dashed lines. For each component of the GMM, the mean is shown as the $+$ and the 3σ covariance ellipse is shown as the solid line.

3 measure the bearing to the target and nodes 4 and 5 measure the range. Therefore, the state of the target is not observable from any single node.

Each node takes a measurement once per second and the probability of detection is 1. When the measurements are first received, each node initializes a GMM to represent the available information. The initialized results are shown in Figure 2. The bearing-only sensor uses the range parameterized Kalman filter proposed by Peach [9]. Peach observed that, for bearings only tracking, a Kalman filter using modified polar Cartesian coordinates is consistent providing

$$\frac{\sigma_R}{R} \leq 0.2.$$

A bank of filters are initialized, each with the same weight, and each using the bearing estimate and a nominal range to initialize the track position. In our experiments, we found that if we used the Unscented Transformation [22] and Cartesian coordinates, we could increase the above inequality to 0.5. To initialize the range-only estimate we used an angle parameterized Kalman filter [8]. This is conceptually very similar to the range-parameterized filter: a set of hypotheses are generated for different nominal values of the bearing. The range data was extracted from the sensor and the bearing covariance was set to be sufficiently large to ensure that adjacent covariance ellipses overlapped. Experiments indicated that a 45° degree spacing, leading to 8 modes, was sufficient to provide stable tracking.

Once a sensor had initialized a target, each mode was predicted and updated locally using the standard GMM update rules [23]. After every 10 time steps the nodes compressed their estimates for distribution and broadcast their estimates to other nodes. The compression step merged the GMM estimates in each mode into a mixture of four components. This step was carried out for two reasons. The first was to reduce the number of parameters which must be distributed between the nodes. Second, by reducing the number of modes, the computational cost of the update algorithm is greatly reduced. We used the Integral Squared Error Reduction algorithm (ISER) developed by Williams and Maybeck [24]. This algorithm uses a greedy approach to merge components such that the integral squared distance between the original distribution, $P(\mathbf{x})$ and the approximate distribution $\hat{P}(\mathbf{x})$,

$$\int \left(P(\mathbf{x}) - \hat{P}(\mathbf{x})\right)^2 d\mathbf{x}$$

is minimized.

Each node broadcast its state estimate to all other nodes. The probability that an update was received was 70%. Furthermore, no acknowledgment scheme was used and so no node knows if an estimate received its communication. Therefore, the topology of the network is, in effect, adhoc and time varying.

If a node received a broadcast estimate, it fused that estimate using one of the update schemes into its local estimate and the number of components were reduced back to 4 (for nodes 1–3) or 8 (for nodes 4–5) using ISER to prevent the combinatorial explosion in the number of terms in the GMM.

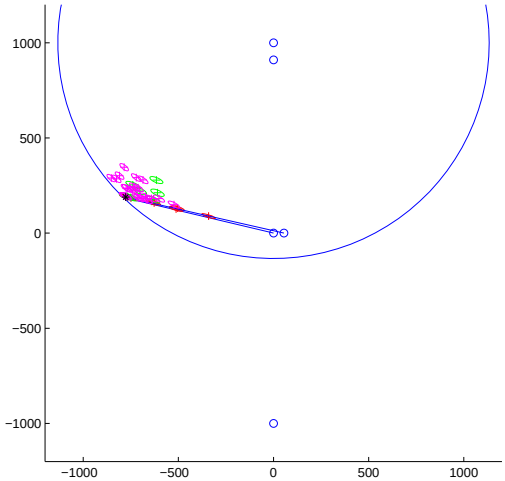
Three algorithms were tested:

1. **Naive Bayes.** This assumes that the estimates are independent. The standard GMM equations are used to fuse local and remote estimates.

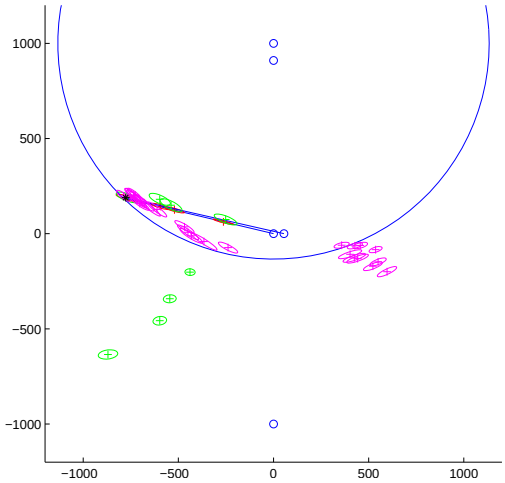
2. **PCCI** This uses the PCCI equations in (7) with the cost function to minimize the determinant of each $\mathbf{C}_{ij}$.

3. **Pseudo-Chernoff**. This uses the pseudo-Chernoff equations in (9) with the cost to minimize $\mathbf{C}$ in (10).

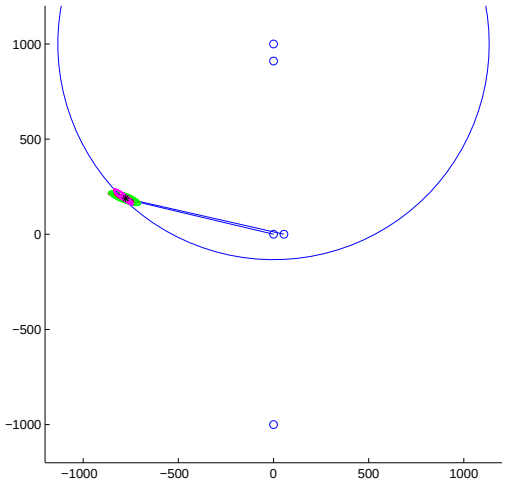
Each algorithm has qualitatively different results as illustrated in Figure 3. This figure plots the components of each estimate for each node at time step 31. At this time step the target has been detected by nodes 1, 2 and 5 and the nodes have just completed a distributed update. In this instance, all nodes received updates from all other nodes. Because both range and bearing data are being fused together, the results should lead to an estimate which is tightly clustered about the intersection of the range and bearing measurements. However, the components of the naive Bayes estimate, shown in Figure 3(a), are scattered around the intersection region. The PCCI algorithm, on the other hand, scatters its components much more widely. As can be seen in Figure 3(b), the estimates lie in two main clusters. The first cluster lies to the left and is near the intersection between the range and bearing estimates. The second cluster lies to the right and is, in fact, behind the range sensor. The results from the pseudo-Chernoff algorithm are shown in Figure 3(c) and, as can be seen, all the components lie at the intersection region.



(a) Naive Bayes.

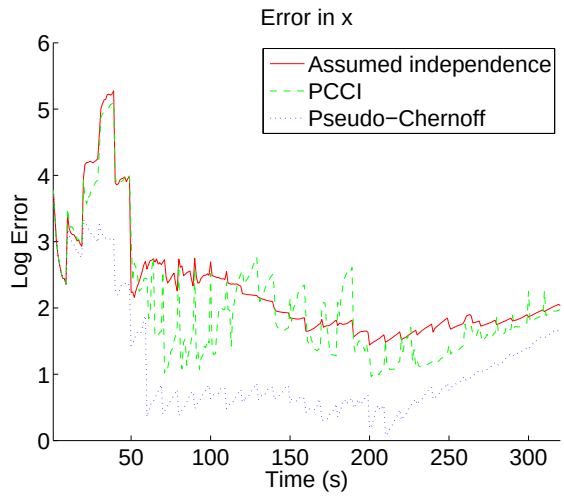


(b) PCCI.

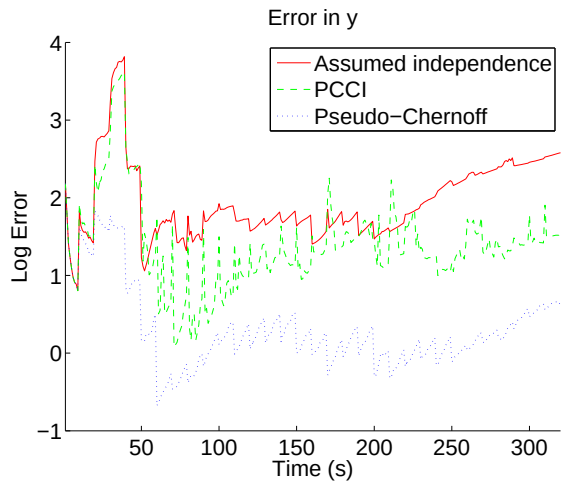


(c) Pseudo-Chernoff

Figure 3: The state of the estimate at time step 30. The location of the true target is $\star$ (most clearly seen in the first figure). The locations of the sensing platforms are $\circ$. The measurements taken by the bearing sensors are shown as lines, those for the range circles as sensors. The 4σ covariance ellipses for each component in each node estimate is plotted.



(a) MSE in x estimate.

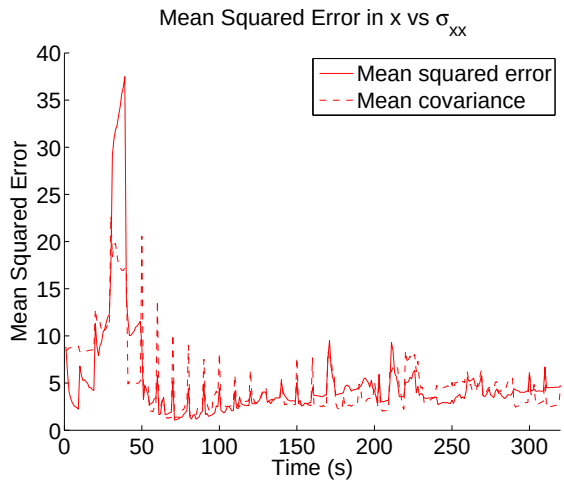


(b) MSE in y estimate.

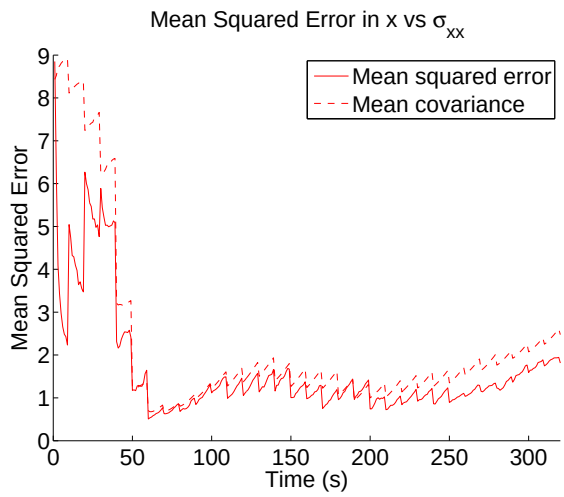
Figure 4: The mean squared errors in the estimates.

The above results illustrate, not surprisingly, that the naive Bayes algorithm leads to an extremely poor approximation to the state estimate. Furthermore, it could be argued that the PCCI is more conservative: rather than place all of the components in one location, it tends to distribute them more. These results are partially confirmed by Figure 4 which plots the mean squared errors in each algorithm. As can be seen, the MSE in the naive Bayes and PCCI are similar to one another. The MSE in the pseudo-Chernoff algorithm is significantly smaller.

However, the apparently conservative nature of the PCCI algorithm is not evident in Figure 5. This figure plots the actual mean squared error in x versus the mean standard deviation (calculated from the covariance matrix) for 100 Monte Carlo runs. As can be seen, the errors in the PCCI algorithm show regular spikes due to the distributed updates. In a number of instances, the true mean squared error is greater than that calculated by the filter. In contrast, the mean squared error in the Pseudo-Chernoff algorithm actually falls at each update step and the true mean squared error is less than that estimated by the filter.



(a) PCCI.



(b) Pseudo-Chernoff.

Figure 5: The log mean squared error estimates and covariances in the x estimates for filter 1, calculated from 100 Monte Carlo runs.

5 Conclusions

This paper has conducted an empirical study into the use of Chernoff information to provide robust algorithms for the fusion of GMMs with unmodified correlations in distributed environments. We have derived a first order approximation which we have shown is consistent and more accurate than the PCCI. These results provide additional evidence that Chernoff information provides a potentially valuable extension of CI to more general classes of probabilistic distributions.

There are several issues to be addressed. First and foremost, it is still unclear what properties are actually guaranteed by the Chernoff Information. The CI algorithm has the property that, providing the conditions in (1) are satisfied, then (2) is satisfied as well. However, it is not clear if an equivalent condition can be specified in the input and output distributions. Second, a more detailed analysis of the effects of the first-order approximation must be carried out. A higher order expansion will, for example, lead to a more accurate estimate. Third, experiments should be conducted to explore the effect of using different, and potentially more accurate, cost functions on the estimate.

Acknowledgments

The authors would like to thank Jason Williams (now at MIT) for providing the code to implement the ISER component reduction algorithm.

References

[1] S. J. Julier and H. F. Durrant-Whyte. On the Role of Process Models in Autonomous Land Vehicle Navigation Systems. *IEEE Transactions on*

Robotics and Automation, 19(1):1–14, February 2003.

[2] Martin E. Liggins II, Chee-Yee Chong, Ivan Kadar, Mark G. Alford, Vincent Vannicola, and Stelios Thomopoulos. Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(2):95–107, 1997.

[3] S. Grime and H. F. Durrant-Whyte. Data fusion in decentralized sensor fusion networks. *Control Engineering Practice*, 2(5):849–863, 1994.

[4] Simukai W. Utete. *Network Management in Decentralised Sensing Systems*. PhD thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, January 1995.

[5] L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh and H. Durrant-Whyte. Decentralised data fusion with particles. In Claude Sammut, editor, *Proceedings of the Australasian Conference on Robotics and Automation 2004*, 5–7 December 2004.

[6] J. K. Uhlmann. *Dynamic Map Building and Localization for Autonomous Vehicles*. PhD thesis, University of Oxford, 1995.

[7] S. J. Julier and J. K. Uhlmann. A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations. In *Proceedings of the IEEE American Control Conference*, volume 4, pages 2369–2373, Albuquerque NM, USA, June 1997.

[8] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[9] N. Peach. Bearings-only tracking using a set of range-parameterised extended kalman filters. *IEE Proceedings – Control Theory and Applications*, 142(1):73–80, January 1995.

[10] R. Mahler. Optimal / robust distributed data fusion: A unified approach. In Ivan Kadar, editor, *The Proceedings of AeroSense: The 14th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, volume 4052 of *Signal Processing, Sensor Fusion, and Target Recognition IX*, pages 128–138, Orlando, FL, USA, August 2000 2000. SPIE, SPIE.

[11] M. Hurley. An information-theoretic justification for covariance intersection and its generalization. In *Proceedings of the 2002 FUSION Conference*, Annapolis, MD, USA, 7–11 July 2002.

[12] B. Upcroft. Rich probabilistic representations for bearing only decentralised data fusion. In *Proceedings of FUSION'05*, Philadelphia, PA, USA, 25–29 July 2005.

[13] A. G. O. Mutambara. *Decentralized Estimation and Control for Nonlinear Systems*. CRC Press, Boca Raton FL, USA, 1998.

[14] T. M. Berg and H. F. Durrant-Whyte. Model Distribution in Decentralized Multi-Sensor Data Fusion. In *Proc. American Control Conference*, volume 3, pages 2292–2293. IEEE, 1991.

[15] C. Chong and S. Mori. Graphical models for nonlinear distributed estimation. In *Proceedings of FUSION '04*, Stockholm, Sweden, 28th June – 1st July 2004.

[16] J. O'Brien. An algorithm for the fusion of correlated probabilities. In *Proceedings of FUSION'98*, volume 2, pages 565–571, Las Vegas, NV, USA, 6–9 July 1998.

[17] I. Hwang, K. Roy, H. Balakrishnan, and C. Tomlin. A distributed multiple-target identity management algorithm in sensor networks. In *The Proceedings of the 43rd IEEE Conference on Decision and Control*, December 2004.

[18] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

[19] R. S. Bucy and K. D. Senne. Digital synthesis of non-linear filters. *Automatica*, 7:287–298, 1971.

[20] P. J. Davis and R. Rabinowitz. *Methods of Numerical Integration*. Academic Press, New York, NY, USA, 1984.

[21] Jacob Goldberger and Hagai Aronowitz. A distance measure between gmms based on the unscented transform and its application to speaker recognition. In *Proceedings of the 9th Eurospeech Conference*, Lisbon, Portugal, 4–8 September 2005.

[22] S. J. Julier, J. K. Uhlmann and H. F. Durrant-Whyte. A new approach for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, March 2000.

[23] D. L. Alspach and H. W. Sorenson. Nonlinear Bayesian Estimation Using Gaussian Sum Approximations. *IEEE Transactions on Automatic Control*, AC–17(4):439–447, August 1972.

[24] Jason L. Williams and Peter S. Maybeck. Cost-function-based hypothesis control techniques for multiple hypothesis tracking. In *Proc. Sixth International Conference of Information Fusion*, pages 1047–1054. International Society of Information Fusion, July 2003.