The Optimization Model of Target Recognition Based on Wireless Sensor Network

Zheng Dou, Yu Sun, and Yun Lin

Harbin Engineering University, No. 145 Nantong Street, Nangang District, Room 443, Building No. 21, Heilongjiang Province, Harbin 150001, China

Correspondence should be addressed to Yun Lin; linyun@hrbeu.edu.cn

Received 11 June 2014; Accepted 2 August 2014; Published 14 August 2014

1. Introduction

With the rapid development of wireless sensor network technology, it is possible to gather large amount of information in real time from a lot of information sources. Such information must be processed appropriately in order to generate an optimal recognition. In order to reduce complexity in the recognition process and to improve effectiveness of the recognition system, the information collected by different sensors is usually first processed in the locality and only some brief reports are transmitted to recognition center. These brief reports usually take the form of preliminary recognition and are typically imprecise, inconsistent, or fuzzy. For example, the first sensor may describe “it is a hostile strategic bomber with probability 1,” while the second sensor may declare “it is a hostile strategic bomber with probability 0.1.” Obviously within a probabilistic framework, the above two statements are inconsistent and fuzzy. Therefore, the real challenge for the recognition center is how to combine these imperfect sensor reports properly and to get a more reasonable recognition.

The common mathematical model of target recognition is based on Bayesian Inference Model and Dempster-Shafer (D-S) Evidential Inference Model [1]. The former model is well developed in statistical recognition theory, but it suffers from the fact that it requires the knowledge of a priori distribution and the conditional probabilities of sensor reports. In contrast, the D-S evidence inference model does not require such a priori knowledge, but it requires the sensor reports to be of a certain restrictive form and to be not inconsistent or fuzzy.

Convex optimization model is ubiquitously used in communication system [2], signal processing [3], and multisensor information fusion [4, 5]. Furthermore, many engineering problems can be converted into convex optimization problems, which greatly facilitate their analytic and numerical solutions. Therefore, in this paper, based on the convex optimization theory, a new optimization model of target recognition is proposed. The basic idea is to choose a probability distribution which provides an overall best “fit” of the potentially inconsistent sensor reports. It converts the target recognition to a convex quadratic programming model.
formulation [6], which has only a polynomial-time growing computational complexity [7, 8]. Therefore, it can be solved very efficiently in real time. In addition, unlike D-S evidence inference model, it can deal with inconsistent or incomplete sensor reports without any restriction. According to the numerical examples, in contrast to D-S evidence inference model, this new optimization model has a lower computational complexity, better recognition result, more widespread applicability, and stronger robustness.

This paper is organized as follows. Section 1 is the introduction. Section 2 provides a simple review of the Bayesian model and D-S evidence inference model. Section 3 describes model with sensor reports by using the convex quadratic formulation [6], which has only a polynomial-time growing complexity. Section 4 gives the solution method based on the Logarithmic Penalty Barrier Function [9, 10]. Section 5 gives several numerical examples in contrast to the D-S evidence inference model. Section 6 describes the conclusion of the paper.

2. The Common Model of Target Recognition

Suppose Ω = {a1, a2, ..., an} is the target recognition framework, which has n mutually exclusive and exhaustive propositions.

2.1. Bayesian Model. The Bayesian model is one of the target recognition methods, which is based on the knowledge of the a priori probability distribution p(a_i) over Ω, and the knowledge of the conditional probabilities of the form P(ω | a_i), which shows the probability of sensor report ω given that the true proposition is a.

These conditional probabilities are assumed to be known and constant. Whenever a sensor report ω is given, the updated probabilities are computed by Bayes’ rule, which is shown as follows:

\[
P(a_i | ω) = \frac{P(ω | a_i) P(a_i)}{P(ω)} ,
\]

where

\[
P(ω) = \sum_ω P(ω | a_i) P(a_i) .
\]

Then P(a_i | ω) becomes the new prior probability in the next update. In other words, Bayes’ rule is solved recursively as new sensor reports arrived. This simple relation describes the recognition center’s estimate of current probabilities given all the sensor reports.

As discussed above, Bayes’s rule requires the value of the prior probabilities p(a_i) and the values of conditional probabilities P(ω | a_i). In practice these values are usually developed using past experience and are difficult to justify.

2.2. Dempster-Shafer (D-S) Evidence Inference Model. Another model of target recognition is D-S evidence inference model, which has been shown to be effective in approximate reasoning and artificial intelligence and is widely used in practice. This model of evidence reasoning by D-S can assign a probability to any of the original n propositions or to disjunctions of the propositions. For example, a union of a_1 and a_2 will have a probability of P(a_1 ∨ a_2). Notice that there are a total of M = 2^n − 1 different possible unions for the propositions in Ω that may be assigned a probability mass. Not all of these unions must be assigned a probability. The sum of all the assigned probabilities must be one. It should be pointed out that this general form of representation is different from the Bayesian approach in which the probabilities are assigned only to the individual propositions, rather than the unions—the probability of a union of propositions is defined as the sum of the probabilities of the individual propositions.

Suppose there are two evidences E_1 and E_2, the corresponding probability masses are m_1 and m_2, the focal elements are X_i and Y_j, and the D-S fusion rule is shown as follows:

\[
m(A) = \begin{cases} \frac{1}{1-t \sum_{X∩Y=E} m_1(X_i) \cdot m_2(Y_j)}, & E \neq 0, \\ 0, & E = 0, \end{cases}
\]

where t is the conflict coefficient between evidences, which is shown as follows:

\[
t = \sum_{X∩Y≠∅} m_1(X_i) \cdot m_2(Y_j) .
\]

Meantime, ∅ represents the empty set, which is contrary to the universal set Ω.

According to the formula above, we can know that, if t = 1, it will have a high degree of evidence conflict; therefore, it needs to improve the D-S fusion rule.

Therefore, the fused probability measure will be regarded as the final target recognition result. However, since there are exponentially M = 2^n − 1 many subsets of Ω, the above recognition rule suffers from exponentially growing complexity. For large values of M, the computation may be difficult to finish in real time.

3. The Convex Optimization Model of Target Recognition

3.1. The Reports of Sensors. For the subsets of propositions ω_k(1), ω_k(2), ..., ω_k(m_k) in Ω, m_k represents the number of subsets in sensor report. This relative likelihood of occurrence is shown by r_k(1), r_k(2), ..., r_k(m_k), r_k(i) ∈ [0, 1], and \( r_k(1) + r_k(2) + \cdots + r_k(m_k) = c_k \) which represents the confidence degree of sensor reports.

In other words, with certainty c_k, the probabilities of ω_k(1), ω_k(2), ..., ω_k(m_k) must satisfy

\[
P_k[ω_k(1)] : P_k[ω_k(2)] : \cdots : P_k[ω_k(m_k)] = r_k(1) : r_k(2) : \cdots : r_k(m_k).
\]

Here k is the index of sensors, m_k is the index of subset, and
$P_k[\omega_k(i)]$ is the occurrence probability of subset $\omega_k(i)$. The confidence degree $c_{rk}$ reflects the certainty in reports, and $c_{rk} \in [0, 1]$. For example, $c_{rk} = 0$ represents the least confidence, and $c_{rk} = 1$ represents the full confidence.

This kind of report form has no restrictions imposed on how the subsets are selected. The selected subsets are not necessarily mutually exhaustive or exclusive.

The following is a simple example; suppose the target recognition framework is $\Omega = \{a_1, a_2, a_3, a_4\}$ and there are three sensors. These reports are given as follows.

The report of sensor 1 is shown as follows:

\[
P_1 = \begin{cases} 
  c_{r1} = r_1(1) + r_1(2) + r_1(3) + r_1(4) = 0.85 \\
  P_1[\omega_1(1)] : P_1[\omega_1(2)] : P_1[\omega_1(3)] : P_1[\omega_1(4)] 
\end{cases}
\]

\[
= r_1(1) : r_1(2) : r_1(3) : r_1(4) = 0.60 : 0.20 : 0.05 : 0.00,
\]

(6)

with $\omega_1(1) = a_1 \lor a_2$, $\omega_1(2) = a_2 \lor a_3$, $\omega_1(3) = a_1 \lor a_4$, $\omega_1(4) = a_1$ and $r_1(1) = 0.6$, $r_1(2) = 0.2$, $r_1(3) = 0.05$, $r_1(4) = 0$.

The report of sensor 2 is shown as follows:

\[
P_2 = \begin{cases} 
  c_{r2} = r_2(1) + r_2(2) + r_2(3) + r_2(4) = 0.80 \\
  P_2[\omega_1(1)] : P_2[\omega_1(2)] : P_2[\omega_1(3)] : P_2[\omega_1(4)] 
\end{cases}
\]

\[
= r_2(1) : r_2(2) : r_2(3) : r_2(4) = 0.10 : 0.40 : 0.20 : 0.10,
\]

(7)

with $\omega_2(1) = a_1 \lor a_2$, $\omega_2(2) = a_1 \lor a_3$, $\omega_2(3) = a_1 \lor a_4$, $\omega_2(4) = a_2 \lor a_4$ and $r_2(1) = 0.1$, $r_2(2) = 0.4$, $r_2(3) = 0.2$, $r_2(4) = 0.1$.

The report of sensor 3 is shown as follows:

\[
P_3 = \begin{cases} 
  c_{r3} = r_3(1) + r_3(2) + r_3(3) + r_3(4) = 0.84 \\
  P_3[\omega_1(1)] : P_3[\omega_1(2)] : P_3[\omega_1(3)] : P_3[\omega_1(4)] 
\end{cases}
\]

\[
= r_3(1) : r_3(2) : r_3(3) : r_3(4) = 0.22 : 0.35 : 0.12 : 0.15,
\]

(8)

with $\omega_3(1) = a_2$, $\omega_3(2) = a_3$, $\omega_3(3) = a_1$, $\omega_3(4) = a_4$ and $r_3(1) = 0.22$, $r_3(2) = 0.35$, $r_3(3) = 0.12$, $r_3(4) = 0.15$.

Furthermore, it should be known how to get the relative likelihood $r_k(i)$ of the subsets $\omega_k(i)$, which is a very important problem. There are many methods to get the $r_k(i)$, such as fuzzy theory [11], grey association theory [12], rough set theory [13], and neural network [14]. However, in this paper, because of the space limitation, we don’t talk about it.

3.2. Optimization Model of Target Recognition. To formulate the problem of target recognition, suppose there are a total of $K$ sensors, which make observations within a surveillance region. Moreover, assume that there are a total of $n$ possible propositions, as specified by $\Omega = \{a_1, a_2, \ldots, a_n\}$, regarding the targets in the surveillance region. Each sensor, based on its own observation and processing, generates several reports which will be sent to the recognition center. These sensor reports can be described as follows:

\[
P_k = \begin{cases} 
  c_{rk} = r_k(1) + r_k(2) + \cdots + r_k(m_k) \\
  P_k[\omega_k(1)] : P_k[\omega_k(2)] : \cdots : P_k[\omega_k(m_k)] 
\end{cases}
\]

(9)

Here $k = 1, 2, \ldots, K$. Based on its own likelihood assessment of each sensor, the recognition center assigns a certain confidence degree of sensor’s report. Let $a_k \in [0, 1]$ represent the recognition center’s confidence assigned to the report from sensor $k$.

Suppose the sensor reports which are shown in formula (9) and the set of fusion sensor’s confidence levels $\{a_k\}$ are given. The goal of target recognition is to determine a set of probabilities:

\[
P(a_1) = p_1, P(a_2) = p_2, \ldots, P(a_n) = p_n,
\]

(10)

for $\Omega$ that best fits the given sensor reports. Clearly, the chosen probabilities must satisfy

\[
p(a_1) + p(a_2) + \cdots + p(a_n) = p_1 + p_2 + \cdots + p_n = 1.
\]

(11)

The basic idea of target recognition is to first set up a cost function for each sensor report and then minimize a weighted sum of all the cost functions subject to the probability constraint which is shown in formula (11). Intuitively, the cost function should measure the difference between $P_k[\omega_k(i)]$ (the sensor’s own estimate of probability distribution) and $P(\omega_k(i))$ (the true probability distribution). In particular, if $P_k[\omega_k(i)] = P(\omega_k(i))$, the cost function should be zero and should increase as $P_k[\omega_k(i)]$ drifts away from $P(\omega_k(i))$.

There are many cost functions which can be used; for example,

\[
C_k(P) = (c_{rk})^2[\sum_{i=1}^{m_k} \left( \frac{r_k(i)}{c_{rk}} - \frac{P(\omega_k(i))}{P_k[\omega_k(1)]} \right)^2, \quad (12)
\]

\[
C_k(P) = \sum_{i=1}^{m_k} \left| c_k(P)r_k(i) - c_{rk}P(\omega_k(i)) \right|, \quad (13)
\]

\[
C_k(P) = \max \left\{ \left| c_k(P)r_k(i) - c_{rk}P(\omega_k(i)) \right| \right\}, \quad (14)
\]

In this paper, the first cost function is used, which is shown in formula (12), with

\[
c_k(P) = P_k[\omega_k(1)] + P_k[\omega_k(2)] + \cdots + P_k[\omega_k(m_k)]. \quad (15)
\]
For the cost function \( C_k(P) \), the term

\[
\sum_{i=1}^{m_k} \left( \frac{r_k(i)}{c_{r,k}} - \frac{P[\omega_k(i)]}{c_k(P)} \right)^2
\]

is the normalized difference in the sensor report as measured by \( P(\omega_k(i)) \) and \( P_k(\omega_k(i)) \). This difference is weighted by \( (c_{r,k})^2 \) to reflect the confidence degree of the sensor report and further by \( [c_k(P)]^2 \) to reflect the likely importance of the report.

If \( P_k(\omega_k(i)) = P(\omega_k(i)) \), the condition \( P_k[\omega_k(1)] : P_k[\omega_k(2)] : \cdots : P_k[\omega_k(m_k)] = r_k(1) : r_k(2) : \cdots : r_k(m_k) \)

implies

\[
\frac{r_k(i)}{c_{r,k}} = \frac{r_k(i)}{r_k(1) + r_k(2) + \cdots + r_k(m_k)} = \frac{P_k[\omega_k(i)]}{P_k[\omega_k(1)] + P_k[\omega_k(2)] + \cdots + P_k[\omega_k(m_k)]}
\]

\[= \frac{P_k[\omega_k(i)]}{c_k(P)}. \quad (17)\]

It further implies \( C_k(P) = 0 \) which proves the effectiveness of cost function. Further simplification of \( C_k(P) \) is shown as follows:

\[
C_k(P) = (c_{r,k})^2[c_k(P)]^2 \sum_{i=1}^{m_k} \left( \frac{r_k(i)}{c_{r,k}} - \frac{P[\omega_k(i)]}{c_k(P)} \right)^2, \quad (18)
\]

which is a convex quadratic programming function of the variable \( P = \{p_1, p_2, \ldots, p_n\} \).

Now the problem of target recognition as the following convex quadratic programming problem can be formulated. Minimize

\[
C(P) = \sum_{k=1}^{K} \alpha_k^2 C_k(P)
\]

\[
= \sum_{k=1}^{K} \alpha_k^2 \sum_{i=1}^{m_k} \left( \frac{r_k(i)}{c_{r,k}} - \frac{P[\omega_k(i)]}{c_k(P)} \right)^2
\]

\[
= \sum_{k=1}^{K} \alpha_k^2 \sum_{i=1}^{m_k} \left[ c_k(P) - c_{r,k} P[\omega_k(i)] \right]^2, \quad (19)
\]

subject to

\[
\sum_{j=1}^{n} p_j = 1; \quad p_j \geq 0, \quad j = 1, 2, \ldots, n. \quad (20)
\]

Here, \( \alpha_k \in [0, 1] \) is reliability degree of the \( k \)th sensor which represents the reliability degree of sensor report and is relative to noise, interference, electromagnetic environment, sensors type, and so on. It can be used to describe the working condition of sensor by considering \( \alpha_k \), it will make the robustness of recognition model better. Furthermore, it can be gotten by a fuzzy neuron network [15].

Furthermore, by decomposing the formula (19), the standard form of convex quadratic programming can be gotten as follows:

\[
\min C(P) = \frac{1}{2} P^T Q P + c^T P + \text{const}
\]

s.t. \( AP = b, \quad P \succeq 0 \). \quad (21)

Here, \( P = (p_1, p_2, \ldots, p_n) \in \mathbb{R}^n \) represents the optimal solution, \( b = (1, 1, \ldots, 1) \in \mathbb{R}^n \), \( A \in \mathbb{R}^{n \times n} \) (all of its elements are one), \( c = (c_1, c_2, \ldots, c_n) \in \mathbb{R}^n \), \( c_j = -2 \sum_{k \in K_j} \alpha_k^2 \sum_{i=1}^{m_j} r_k(i) c_{r,k} u_{k,j,i} \), const = \( \sum_{k \in K_j} \alpha_k^2 \sum_{i=1}^{m_j} r_k(i)^2 \); \( Q = Q_1 + Q_2 \) is a positive semidefinite matrix which is shown as follows:

\[
Q_1 = \left( \begin{array}{cccc}
2 \sum_{k \in K_1} \alpha_k^2 \sum_{i=1}^{m_1} (c_{r,k})^2 (u_{k,1,1})^2 & 2 \sum_{k \in K_1} \alpha_k^2 \sum_{i=1}^{m_1} (c_{r,k})^2 (u_{k,1,2})^2 & \cdots & 2 \sum_{k \in K_1} \alpha_k^2 \sum_{i=1}^{m_1} (c_{r,k})^2 (u_{k,1,n})^2 \\
2 \sum_{k \in K_1} \alpha_k^2 \sum_{i=1}^{m_1} (c_{r,k})^2 (u_{k,2,1})^2 & 2 \sum_{k \in K_1} \alpha_k^2 \sum_{i=1}^{m_1} (c_{r,k})^2 (u_{k,2,2})^2 & \cdots & 2 \sum_{k \in K_1} \alpha_k^2 \sum_{i=1}^{m_1} (c_{r,k})^2 (u_{k,2,n})^2 \\
& \ddots & \ddots & \ddots \\
2 \sum_{k \in K_1} \alpha_k^2 \sum_{i=1}^{m_1} (c_{r,k})^2 (u_{k,n,1})^2 & 2 \sum_{k \in K_1} \alpha_k^2 \sum_{i=1}^{m_1} (c_{r,k})^2 (u_{k,n,2})^2 & \cdots & 2 \sum_{k \in K_1} \alpha_k^2 \sum_{i=1}^{m_1} (c_{r,k})^2 (u_{k,n,n})^2 \\
\end{array} \right)
\]
Suppose \( i \) tc a nb ep r o v e dt h a ti f (24) is converging to the optimal solution of formula (21).

The above linearly constrained convex quadratic program-

\[ Q_2 = \begin{pmatrix}
2 \sum_{k \in K_2} \alpha_k \sum_{i=1}^{m_k} (y_{k,i,1})^2 & 2 \sum_{k \in K_2} \alpha_k \sum_{i=1}^{m_k} y_{k,i,1} y_{k,i,2} & \cdots & 2 \sum_{k \in K_2} \alpha_k \sum_{i=1}^{m_k} y_{k,i,1} y_{k,i,n} \\
2 \sum_{k \in K_2} \alpha_k \sum_{i=1}^{m_k} y_{k,i,2} y_{k,i,1} & 2 \sum_{k \in K_2} \alpha_k \sum_{i=1}^{m_k} (y_{k,i,2})^2 & \cdots & 2 \sum_{k \in K_2} \alpha_k \sum_{i=1}^{m_k} y_{k,i,2} y_{k,i,n} \\
\vdots & \vdots & \ddots & \vdots \\
2 \sum_{k \in K_2} \alpha_k \sum_{i=1}^{m_k} y_{k,i,n} y_{k,i,1} & 2 \sum_{k \in K_2} \alpha_k \sum_{i=1}^{m_k} y_{k,i,n} y_{k,i,2} & \cdots & 2 \sum_{k \in K_2} \alpha_k \sum_{i=1}^{m_k} y_{k,i,n} y_{k,i,n}
\end{pmatrix}
\]

(22)

Here

\[ v_{k,i,j} = r_k(s) \sum_{j=1}^{m_k} u_{k,i,j} - \sum_{j=1}^{m_k} u_{k,i,n} \]

\[ K_1 = \{ k \mid (k = 1, 2, \ldots, K) \]
\[ \cap (u_{k,i,1} = u_{k,i,2} = \cdots = u_{k,i,n} = 1) \}

\[ K_2 = \{ k \mid (k = 1, 2, \ldots, K) \]
\[ \cap \left[ (1 - u_{k,j,j})^2 + (1 - u_{k,j,2})^2 \right. \]
\[ + \cdots + (1 - u_{k,j,n})^2 \neq 0 \} \]

and if the \( j \)th proposition of \( \Omega \) consists in the \( i \)th subset of the \( k \)th sensor, \( u_{k,i,j} = 1 \); otherwise \( u_{k,i,j} = 0 \). Therefore, according to each sensor report, the recognition center can get the \( Q \) and it. Furthermore, the problem of target recognition can be converted to a convex quadratic programming problem. Finally, we can calculate the optimal solution \( P \) as the final recognition result.

4. The Solution Method of Optimization Model

The above linearly constrained convex quadratic programming problem has \( n \) variables with a simplex constraint. Furthermore, in contrast to D-S evidence inference model which has exponential time complexity, the new optimization model has only polynomial time complexity. Therefore, it can be solved very efficiently. In this paper, the solution method of Logarithmic Penalty Barrier Function is introduced to get the optimal solution \( P \). The detailed solution and steps are described as follows [16].

Suppose \( \mu \) is the penalty factor and \( \log p_i \) is the barrier function according to the method of Logarithmic Penalty Barrier Function, the problem (21) can be converted to solve the following problem:

\[ \min \ \varphi(P, u) = \frac{1}{2} P^T Q P + c^T P - u \sum_{i=1}^{n} \log p_i \]
\[ \text{s.t.} \ AP = b, \ P > 0 \]  

(24)

It can be proved that if \( u \to 0 \), then the solution of formula (24) is converging to the optimal solution of formula (21). Suppose \( \Omega_{int} = \{ AP = b, P > 0 \} \); \( P \) is the feasibility interior point of the formula (21), which is \( P \in \Omega_{int} \neq 0 \). Here, 0 represents the empty set, which is contrary to the solution set \( \Omega_{int} \). According to the initial interior point \( P^i \), \( P = P^k \in \Omega_{int} \) can be calculated which is the \( k \)th iteration result. Now, the iteration factor \( d^k \) can be calculated by the following formula:

\[ \min \ \nabla B^T d + \frac{1}{2} d^T (\nabla^2 B) d \]
\[ \text{s.t.} \ AD = 0 \]

(25)

with

\[ \nabla B = \nabla B(P, u) = C + QP - uP^{-1} e \]
\[ \nabla^2 B = \nabla^2 B(P, u) = Q + uP^{-2}; \]

\[ P = \text{Diag}(p_1, p_2, \ldots, p_n), e = (1, 1, \ldots, 1)^T \in \mathbb{R}^n \].

Furthermore, the solution method of formula (25) can be converted to solve the formula

\[ \begin{bmatrix} \nabla^2 B & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d \\ -\lambda \end{bmatrix} = - \begin{bmatrix} \nabla B \\ 0 \end{bmatrix}. \]

(27)

Here, \( \lambda \in \mathbb{R}^m \) is the vector of Lagrange in formula (25).

If the solution of formula (27) satisfies \( d^k \neq 0 \), then \( d^k \) is the iteration factor of formula (24). In order to ensure the strict feasibleness of iteration solution \( P^{k+1} \), it must choose the proper iteration step. Meantime, it must satisfy

\[ P^{k+1} = P^k + \alpha_k d^k > 0 \]

(28)

In this paper, the iteration steps satisfy, where \( \tau \in (0, 1), \)

\[ \alpha_j = \tau \min \left\{ 1, -\frac{p_j^k}{d_j^k} \mid d_j^k < 0 \right\}, \quad j = 1, \ldots, n \]

(29)

In sum, the steps of solution method for formula (21) are shown as follows.

(1) Choosing the proper parameter, including \( \varepsilon > 0, \mu_i \in (0, 1), \sigma \in (0, 1/2), \beta \in (0, 1), k = 1 \). Furthermore, it should determine the initial interior point \( (P^1) \in \Omega_{int} \neq 0 \). It is a very important step, which will affect the convergence rate and the accuracy of solution.

(2) According to the \( P^k \) and formula (27), the iteration factor \( d^k \) can be calculated. If it satisfies \( \|d^k\| < \varepsilon \), then it stops and jumps out the iteration process. At the moment, \( P = P^k \) is the optimal solution of formula (21). Otherwise, turn to step (3).

(3) According to formula (29), the iteration step (or \( \alpha_k \)) can be calculated, and if it satisfies

\[ B(P^k, u^k) - B(P^k + \alpha_k d^k, u^k) \geq -\sigma \alpha_k(d^k)^T \nabla B(P^k, u^k), \]

(30)
Table 1: The recognition model of numeric simulation one.

| Recognition model          | Recognition result |
|---------------------------|--------------------|
| Evidence inference model [4] | 0.1859 0.0430 0.7712 0 |
| Yager model [17]          | 0.0195 0.0045 0.0810 0 |
| Sets model [18]           | 0.1432 0.0784 0.2745 0.0535 |
| Takahiko model [19]       | 0.2782 0.1662 0.4352 0.1204 |
| Murphy model [20]         | 0.2685 0.1533 0.4705 0.1077 |
| Optimization model        | 0.2759 0.1643 0.4328 0.1270 |

Table 2: The recognition result of numeric simulation two.

| Recognition model          | Recognition result |
|---------------------------|--------------------|
| Evidence inference model [4] | 0.5922 0.1337 0.2742 0 |
| Yager model [17]          | 0.0346 0.0078 0.0160 0 |
| Sets model [18]           | 0.1982 0.0999 0.1519 0.0516 |
| Takahiko model [19]       | 0.3676 0.2068 0.3052 0.1204 |
| Murphy model [20]         | 0.3808 0.2025 0.3034 0.1133 |
| Optimization model        | 0.2871 0.1760 0.3818 0.1551 |

then turn to step (4). Otherwise, choose \( \alpha^k = \alpha^k / 2 \), until it satisfies formula (30).

(4) Getting the iteration solution with the following formula:

\[
P^{k+1} = P^k + \alpha^k d^k, \quad u^{k+1} = \beta u^k.
\]

(31)

(5) Preparing for next iteration process. Make \( k = k + 1 \) and turn to step (2).

5. Numerical Simulation and Analysis

Suppose target recognition framework is \( \Omega = \{a_1, a_2, a_3, a_4\} \); there are three sensors which will send sensor reports to the recognition center. In recognition center, every sensor report will be fused with convex quadratic programming model and gets the recognition result by the method of Logarithmic Penalty Barrier Function. The main parameter of solution is shown as follows:

\[
\varepsilon = 10^{-7}, \quad \mu_1 = 0.7, \quad \sigma = 0.3355,
\]

\[
\tau = 0.9995, \quad \beta = 0.15, \quad P^1 = [0.4, 0.3, 0.2, 0.1]^T.
\]

(32)

(1) Numeric Simulation Example One. Suppose \( \alpha_1 = \alpha_3 = \alpha_5 = 1 \); it means every single sensor is working well. Assume each sensor files one report to the recognition center which are given as follows.

Report from Sensor 1:

\[
c_{r,1} = r_1 (1) + r_1 (2) + r_1 (3) + r_1 (4) = 0.80
\]

\[
P_1 [\omega_1 (1) : P_1 [\omega_1 (2) : P_1 [\omega_1 (3) : P_1 [\omega_1 (4)]]]
\]

\[
= r_1 (1) : r_1 (2) : r_1 (3) : r_1 (4)
\]

\[
= 0.15 : 0.15 : 0.30 : 0.20.
\]

Report from Sensor 2:

\[
c_{r,2} = r_2 (1) + r_2 (2) + r_2 (3) + r_2 (4) = 0.90
\]

\[
P_2 [\omega_2 (1) : P_2 [\omega_2 (2) : P_2 [\omega_2 (3) : P_2 [\omega_2 (4)]]]
\]

\[
= r_2 (1) : r_2 (2) : r_2 (3) : r_2 (4)
\]

\[
= 0.27 : 0.13 : 0.40 : 0.10.
\]

Report from Sensor 3:

\[
c_{r,3} = r_3 (1) + r_3 (2) + r_3 (3) + r_3 (4) = 0.72
\]

\[
P_3 [\omega_3 (1) : P_3 [\omega_3 (2) : P_3 [\omega_3 (3) : P_3 [\omega_3 (4)]]]
\]

\[
= r_3 (1) : r_3 (2) : r_3 (3) : r_3 (4)
\]

\[
= 0.25 : 0.12 : 0.35 : 0.00.
\]

Here, \( \omega_i (m) = a_m, i = 1, 2, 3, m = 1, 2, 3, 4 \).

The recognition result is shown in Table 1. In order to make much better comparison, five different D-S evidence inference models are selected. According to Table 1, convex quadratic programming model has the same recognition
result with D-S evidence inference model; the recognition result is \( a_3 \). Therefore, it is accurate and effective.

(2) Numeric Simulation Example Two. Immediately following the above simulation, suppose, because of the distractions, the third sensor report is changed, which is shown as follows. Meantime, the reliability degree of this sensor report is \( a_3 = 0.6 \).

Report from Sensor 3:

\[
\begin{align*}
c_{r,3} &= r_3 (1) + r_3 (2) + r_3 (3) + r_3 (4) = 0.52 \\
P_3 [\omega_3 (1)] : P_3 [\omega_3 (2)] : P_3 [\omega_3 (3)] : P_3 [\omega_3 (4)] &= r_3 (1) : r_3 (2) : r_3 (3) : r_3 (4) \\
&= 0.32 : 0.15 : 0.05 : 0.00.
\end{align*}
\]

According to Table 2, if the sensor is disturbed, the recognition center will receive wrong reports. D-S evidence inference model will lead to the wrong recognition result. However, convex quadratic programming model takes the reliability degree of every single sensor into account; therefore it can still get the right recognition result of \( a_3 \), which proves that convex optimization model has stronger anti-interference and robustness than that of D-S evidence inference model.

(3) Numeric Simulation Example Three. Suppose \( \alpha_1 = \alpha_2 = \alpha_3 = 1 \), but the sensor reports have high confliction, which are shown as follows. The recognition result is shown in Table 3.

Report from Sensor 1:

\[
\begin{align*}
c_{r,1} &= r_1 (1) + r_1 (2) + r_1 (3) + r_1 (4) = 1 \\
P_1 [\omega_1 (1)] : P_1 [\omega_1 (2)] : P_1 [\omega_1 (3)] : P_1 [\omega_1 (4)] &= r_1 (1) : r_1 (2) : r_1 (3) : r_1 (4) \\
&= 0.98 : 0.01 : 0.01 : 0.
\end{align*}
\]

Report from Sensor 2:

\[
\begin{align*}
c_{r,2} &= r_2 (1) + r_2 (2) + r_2 (3) + r_2 (4) = 1 \\
P_2 [\omega_2 (1)] : P_2 [\omega_2 (2)] : P_2 [\omega_2 (3)] : P_2 [\omega_2 (4)] &= r_2 (1) : r_2 (2) : r_2 (3) : r_2 (4) \\
&= 0 : 0.01 : 0.99 : 0.
\end{align*}
\]

Report from Sensor 3:

\[
\begin{align*}
c_{r,3} &= r_3 (1) + r_3 (2) + r_3 (3) + r_3 (4) = 0.52 \\
P_3 [\omega_3 (1)] : P_3 [\omega_3 (2)] : P_3 [\omega_3 (3)] : P_3 [\omega_3 (4)] &= r_3 (1) : r_3 (2) : r_3 (3) : r_3 (4) \\
&= 0.32 : 0.15 : 0.05 : 0.00).
\end{align*}
\]

According to Table 2, if the sensor is disturbed, the recognition center will receive wrong reports. D-S evidence inference model will lead to the wrong recognition result. However, convex quadratic programming model takes the reliability degree of every single sensor into account; therefore it can still get the right recognition result of \( a_3 \), which proves that convex optimization model has stronger anti-interference and robustness than that of D-S evidence inference model.

Table 3: The recognition result of numeric simulation three.

| Recognition model | \( a_1 \) | \( a_2 \) | \( a_3 \) | \( a_4 \) |
|-------------------|---------|---------|---------|---------|
| Evidence inference model [4] | 0       | 0       | 1       | 0       |
| Yager model [17] | 0       | 0       | 0.001   | 0       |
| Sets model [18] | 0.3206  | 0.0034  | 0.1886  | 0       |
| Takahiko model [19] | 0.6267  | 0.0067  | 0.3676  | 0       |
| Murphy model [20] | 0.6260  | 0.0067  | 0.3673  | 0       |
| Optimization model | 0.6270  | 0.007   | 0.367   | 0       |

Table 4: The recognition result of numeric simulation four.

| Recognition model | \( a_1 \) | \( a_2 \) | \( a_3 \) | \( a_4 \) |
|-------------------|---------|---------|---------|---------|
| Optimization model | 0.0441  | 0.1032  | 0.6038  | 0.2489  |

Report from Sensor 3:

\[
\begin{align*}
c_{r,3} &= r_3 (1) + r_3 (2) + r_3 (3) + r_3 (4) = 1 \\
P_3 [\omega_3 (1)] : P_3 [\omega_3 (2)] : P_3 [\omega_3 (3)] : P_3 [\omega_3 (4)] &= r_3 (1) : r_3 (2) : r_3 (3) : r_3 (4) \\
&= 0.90 : 0.10 : 0.
\end{align*}
\]

According to Table 3, the D-S evidence inference model cannot effectively fuse with such high conflict sensor reports. Yager method and Sets method will make unreasonable recognition result. Takahiko method and Murphy method can get reasonable recognition result, but they need much more complex processes. Convex quadratic programming model has the same recognition result with Takahiko method and Murphy method without complex processes. Therefore, it can deal with high conflict sensor report in real time.

(4) Numeric Simulation Example Four. Suppose \( \alpha_1 = \alpha_2 = \alpha_3 = 1 \), but the subsets of propositions \( \omega_k (1), \omega_k (2), \ldots, \omega_k (m_k) \) are not mutually exclusive or exhaustive, which are shown as follows. The recognition result is shown in Table 4.

Report from Sensor 1:

\[
\begin{align*}
c_{r,1} &= r_1 (1) + r_1 (2) + r_1 (3) + r_1 (4) = 0.85 \\
P_1 [\omega_1 (1)] : P_1 [\omega_1 (2)] : P_1 [\omega_1 (3)] : P_1 [\omega_1 (4)] &= r_1 (1) : r_1 (2) : r_1 (3) : r_1 (4) \\
&= 0.60 : 0.20 : 0.05 : 0.00.
\end{align*}
\]

with \( \omega_1 (1) = a_1 \lor a_3, \omega_1 (2) = a_2 \lor a_4, \omega_1 (3) = a_1 \lor a_4, \omega_1 (4) = a_1 \).
Report from Sensor 2:
\[ c_{2,2} = r_2(1) + r_2(2) + r_2(3) + r_2(4) = 0.80 \]
\[ P_2[\omega_2(1)] : P_2[\omega_2(2)] : P_2[\omega_2(3)] : P_2[\omega_2(4)] = r_2(1) : r_2(2) : r_2(3) : r_2(4) \]
\[ = 0.10 : 0.40 : 0.20 : 0.10, \]
with \( \omega_2(1) = a_1 \lor a_2, \omega_2(2) = a_3 \lor a_4, \omega_2(3) = a_1 \lor a_3, \omega_2(4) = a_2 \lor a_1 \).

Report from Sensor 3:
\[ c_{3,3} = r_3(1) + r_3(2) + r_3(3) + r_3(4) = 0.84 \]
\[ P_3[\omega_3(1)] : P_3[\omega_3(2)] : P_3[\omega_3(3)] : P_3[\omega_3(4)] = r_3(1) : r_3(2) : r_3(3) : r_3(4) \]
\[ = 0.22 : 0.35 : 0.12 : 0.15, \]
with \( \omega_3(1) = a_2, \omega_3(2) = a_3, \omega_3(3) = a_1, \omega_3(4) = a_4 \).

As well known, the D-S evidence inference model needs strict form of sensor reports; the subsets of propositions \( \omega_k(1), \omega_k(2), \ldots, \omega_k(m_k) \) must be mutually exclusive or exhaustive. Therefore, without additional processes, the D-S evidence inference model cannot deal with this kind of sensor reports. However, the convex quadratic programming model can deal with it. Therefore, the convex quadratic programming model has more wide applicability than the D-S evidence inference model.

6. Conclusion

In this paper, a new optimization model of target recognition is proposed. According to the theory analysis and numeric simulation, this new optimization model has the similar recognition performance with the D-S evidence inference model. In addition, in contrast to the D-S evidence inference model, it has some additional advantages. For example, in the optimization model, the reliability degree of all sensors is considered; therefore, it has much stronger anti-interference and robustness. Meantime, it will have much better accuracy and stability in complex noise and interference environment. In addition, the D-S evidence inference model needs to consider all possible subsets of propositions, while this new optimization model only needs to assign probabilities to each individual proposition. Finally, this new optimization model can deal with high conflict sensor reports without additional processes and has only polynomial computation complexity.

According to the discussion in this paper, in order to use the new model in practice, there are three questions needed to be solved in the future. The first question is how to get the sensor reports in different application environment. The second question is how to get the confidence degree of all sensors which can best fit to the real situation. The third question is how to further decrease the computation amount of solution method.

Conflict of Interests

Meantime, all the authors declare that there is no conflict of interests regarding the publication of this article.

Acknowledgments

This work was supported by the Nation Nature Science Foundation of China (no. 61301095 and no. 6120237), Nature Science Foundation of Heilongjiang Province of China (no. QC2012C069 and F201408), and the Fundamental Research Funds for the Central Universities (no. HEUCF1408).

References

[1] E. Waltz and J. Lilnas, Multisensor Data Fusion, Artech House, Boston, Mass, USA, 1990.
[2] M. Chiang, “Geometric Programming for communication systems,” Foundations and Trends in Communications and Information Theory, vol. 2, no. 1-2, pp. 1-156, 2005.
[3] Z. Luo and W. Yu, “An introduction to convex optimization for communications and signal processing,” IEEE Journal on Selected Areas in Communications, vol. 24, no. 8, pp. 1426–1438, 2006.
[4] Y. Lin, X.-C. Si, H. Yang, and Y.-B. Li, “Multisensor target identification technology based on convex optimization theory,” Journal of Harbin Engineering University, vol. 31, no. 4, pp. 513–518, 2010.
[5] N. Li, Z. Luo, K. Max Wong, and E. Bossé, “Convex optimization approach to identity fusion for multisensor target tracking,” IEEE Transactions on Systems, Man, and Cybernetics A. Systems and Humans., vol. 31, no. 3, pp. 172–178, 2001.
[6] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[7] D. Bertsimas and X. Luo, “On the worst case complexity of potential reduction algorithms for linear programming,” Mathematical Programming, vol. 77, no. 3, pp. 321–333, 1997.
[8] A. M. Ben-Amram, N. D. Jones, and L. Kristiansen, “Linear, polynomial or exponential? Complexity inference in polynomial time,” in Logic and Theory of Algorithms, vol. 5028 of Lecture Notes in Computer Science, pp. 67–76, 2008.
[9] J. B. Lasserre, “Why the logarithmic barrier function in convex and linear programming?”, Operations Research Letters, vol. 27, no. 4, pp. 149–152, 2000.
[10] F. Bai and L. Zhang, “Approximate global exact barrier method for logarithmic barrier function,” OR Transactions, vol. 4, no. 3, pp. 13–18, 2000.
[11] N. Li, C. Qu, F. Su, and D. Ping, “Improved radar emitter fuzzy identification algorithm,” Journal of the University of Electronic Science and Technology of China, vol. 39, no. 2, pp. 182–185, 2010.
[12] N. Li, C. Qu, F. Su, and D. Ping, “Application of gray association theory in emitter recognition,” Journal of System Simulation, vol. 21, no. 4, pp. 7896–7898, 2009.
[13] Z. Zheng-chao, G. Xin, H. E. You, L. I. Ying-sheng, and G. U. O. Wei feng, “Study on radar emitter signal recognition based on rough sets and grey association theory,” Jisuan ji yu xian dai hua, vol. 4, pp. 1–5, 2010.
[14] A. Hossen, F. Al-Wadahi, and J. A. Jervase, “Classification of modulation signals using statistical signal characterization and artificial neural networks,” Engineering Applications of Artificial Intelligence, vol. 20, no. 4, pp. 463–472, 2007.
[15] H. Zong, C. Zong, C. Yu, and T. Quan, “Research and application on NFE model of multi-sensor information fusion,” *Journal of Electronics and Information Technology*, vol. 32, no. 3, pp. 522–527, 2010.

[16] R. Roy, “Numerical comparison of different penalty modified barrier functions for optical tomography problems,” in *Progress in Biomedical Optics and Imaging*, vol. 7174 of *Proceedings of SPIE*, 2009.

[17] R. R. Yager, “On the Dempster-Shafer framework and new combination rules,” *Information Sciences*, vol. 41, no. 2, pp. 93–137, 1987.

[18] P. Smets, “Combination of evidence in the transferable belief model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 447–458, 1990.

[19] T. Horiuchi, “Decision rule for pattern classification by integrating interval feature values,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 440–448, 1998.

[20] C. K. Murphy, “Combining belief functions when evidence conflicts,” *Decision Support Systems*, vol. 29, no. 1, pp. 1–9, 2000.