Task prioritization

Ronnie Johansson

26th April 2004

Abstract

Abstract

1 Introduction

Prioritization of information acquisition tasks (or information tasks for short) realizes the *focus of attention* function of our system, i.e., the information tasks with the highest priority (according to the prioritization assignment described below) are in focus. Here we present an intuitively appealing method to calculate the priorities of tasks.[1] The tasks all involve determining the state (in this case the position and direction) of a particular (military) unit. The priority measure we assign to a task depends on both the *estimated threat* posed by the enemy unit that corresponds to the task as well as the *estimated improvement potential*. The prioritization focuses on describing the urgency of tasks without considering how or what resources can satisfy them.

Overview . . . .

2 System view and goals

The overall goal of the system is *plan recognition*, i.e., estimating the current plans or activities of observed enemy companies. In our scenario, the system is a friendly platoon (or, possibly, a set of friendly platoons) that estimates the properties of enemy units with respect to itself. Properties do not only include attributes such as position and direction, but also intentions. For each unit, a task may be created that requests position and direction estimates of the unit. The creation of tasks reflects the need of the system (or some part of the system) to improve its estimation of a corresponding unit. Acquired information (estimates with error covariance) is fused to estimate the current plan (e.g., attack, march, defense, etc) of the unit. The estimation process involves utilizing a *Dynamic Bayesian Network* (DBN) model which produces a probability distribution over plausible enemy plans. The DBN requires information about an enemy position estimate and terrain properties.

The (sensing) resources available (e.g., UAVs and human observers) are, furthermore, in general not sufficient to treat all tasks and some prioritization

[1]From now on, in this document, all information acquisition tasks are referred to simply as tasks.

of tasks (i.e., focus of attention) is required. Hence, there is a conflict between tasks which we settle through prioritization.

To calculate priorities, we integrate the estimated threat posed by the enemy unit (Section 4) with the sensitivity of the estimation (Section 3). The following sections will deal with these two parts in detail.

3 Plan and threat

The fusion process generates plan estimates for all (observed) companies (and their platoon components). We let "plan" to denote the current intentions of the enemy unit. A plan estimate is a probability distribution, $P(a)$, over all (say k) discrete and finite possible plans $a \in \mathcal{A}$. In our case, the probability function $P_A(a)$ is inferred through a dynamic Bayesian network based on fused observations of platoon state $\hat{\mathbf{x}}$. previously inferred plans, and terrain characteristics. The estimate $\hat{\mathbf{x}}$ is composed of the platoon position ($\hat{\mathbf{p}} = (\hat{x}_{coord}, \hat{y}_{coord})$) and direction ($\hat{d}$) estimates.

For convenience, assume that the probability function $P_A(a)$ is defined in this way:

$$P_A(a) = \begin{cases} a_1^*, & \text{if } a = a_1 \\ \vdots & \vdots \\ a_k^*, & \text{if } a = a_k \end{cases}, a_i \in \mathcal{A} \tag{1}$$

The vector of length k, $\mathbf{a}^*$, stores for each element a_i^* the corresponding probability $P_A(a_i)$.

Given the estimated plans (i.e., generated $P_A(\cdot)$ functions) for all enemy platoons one could estimate the *threat* posed by the observed platoons. E.g., naturally, a unit that we expect to be attempting to attack us, poses a greater threat to us than another unit that is merely marching somewhere. However, we introduce two more factors that should be considered when estimating threats:

Impact E.g, what will be the consequences of an attack by a particular unit. Will the expected (perhaps simulated) consequence be more severe than we can afford? Impact may also have other qualities; e.g., one might consider the impact of being noticed by a marching platoon.

Temporal separation E.g., if two units, equal in strength, are expected to attack, the one closer to us (here, "closeness" is expressed in expected travel time or temporal separation) should be considered to be a greater threat than the other, more distant, unit.

Hence, we propose a threat function that integrates the three mentioned factors: plan, impact and temporal separation. The threat function might, for instance, be a linear, weighted summation over the elements of the plan estimate, emphasizing threatening plans:

$$t\big(P_A(a), imp, ts; \mathbf{w}\big) \triangleq \sum_{a \in A} w(a) P_A(a) + w(imp) \cdot imp + w(ts) \cdot ts, \tag{2}$$

where $w(\cdot)$ is the weight function that emphasizes crucial plans and assigns weights to expected impact and temporal separation.

Using the notion of $\mathbf{a}^*$, we can express Equation 2 in vector notation as $t = t(\mathbf{a}^*, \mathbf{w}) = \mathbf{w}{\cdot}\mathbf{a}^{*T}$ (where $\mathbf{w}$ is the vector of weights corresponding to the previously mentioned weight function $w(a)$).

Attached to the estimated unit state is an uncertainty distribution, here assumed to be the Gaussian $p_X(x){\equiv}N(\mathbf{x}; \mu_{\mathbf{x}}, \mathbf{P})$, where $\mu_{\mathbf{x}}$ is the expected value of $\mathbf{x}$ (approximated with $\hat{\mathbf{x}}$) and $\mathbf{P}$ the covariance (approximated with $\hat{\mathbf{P}}$).

As indicated the threat estimate of a unit $P_A(a)$ is actually dependent on the state estimate $\hat{\mathbf{x}}$, i.e., $P_A(a; \hat{\mathbf{x}})$ or $\mathbf{a}^*(\hat{\mathbf{x}})$, a state which is inherently uncertain. It is more costly (computationally), but more robust to instead estimate the expected threat $\bar{t}{\approx}\mu_t$. In order not to clutter up the remaining calculations of this section, we have neglected the impact and temporal separation factors in the equations.

$$\mu_t = \mathbb{E}_{p_X}[t(\mathbf{X})] = \int_{\mathbf{x}} p_X(\mathbf{x})t(\mathbf{x})d\mathbf{x} = \mathbf{w}\int_{\mathbf{x}} N(\mathbf{x}; \hat{\mathbf{x}}, \hat{\mathbf{P}})\mathbf{a}^*(\mathbf{x})d\mathbf{x} \tag{3}$$

The expected value in Equation 3 can not easily be calculated analytically due the complex estimation of $\mathbf{a}^*(\mathbf{x})$. Hence, an approximative method such as *Monte Carlo simulation* is recommended. In general terms, the N-sample Monte Carlo estimate of the expected value of $g(X)$ is

$$\widetilde{g_N} \triangleq \frac{1}{N}\sum_{i=1}^{N} g(x_i). \tag{4}$$

Hence, in our case, with $g(X)$ is replaced with $t(\mathbf{X})$ and

$$\mu_t \approx \widetilde{t_N} = \frac{1}{N}\sum_{i=1}^{N} t(\mathbf{x_i}) = \frac{\mathbf{w}}{N}\sum_{i=1}^{N} \mathbf{a}^*(\mathbf{x_i})^T = \frac{\mathbf{w}}{N}\sum_{i=1}^{N} \begin{bmatrix} a_1^*(\mathbf{x_i}) \\ \vdots \\ a_k^*(\mathbf{x_i}) \end{bmatrix}. \tag{5}$$

We accept $\widetilde{t_N}$ as the estimate $\bar{t}$. Note that $a_l^*(x_i)$ in Equation 5 should be interpreted as the lth component of the vector $a^*(x_i)$ determined by x_i. The notation does <u>not</u> imply that the lth component of a^* could be calculated independently of the other components.

4 Threat sensitivity

The threat sensitivity $s(\cdot)$ is characterized by the variance of the threat function $t(p)$ (which indicates the "stability" of the threat) with respect to plausible unit states. Hence,

$$s(t) \triangleq f\big(\mathbb{V}[t]\big) = f\big(\mathbb{E}[(t-\mu_t)^2]\big) = f\big(\mathbb{E}[t^2(\mathbf{X})] - \mu_t^2\big), \tag{6}$$

for some function $f(\cdot)$ that maps the variance into some custom sensitivity metric. Also in this case, Equation 6 may be difficult to calculate.

We have accepted an estimate for μ_t which was derived in Equation 5. The other expectancy value can be approximated in a similar way (as in Section 3, we leave out the impact and temporal separation factors from the calculation):

$$\mathbb{E}[t^2(\mathbf{X})] \approx \tfrac{1}{N}\textstyle\sum_{i=1}^{N} t^2(\mathbf{x}) = \{\text{Equation 2}\} = \tfrac{1}{N}\textstyle\sum_{i=1}^{N} \Big(\mathbf{w}{\cdot}\mathbf{a}^*(\mathbf{x_i})^T\Big)^2 \tag{7}$$

Unfortunately, we cannot further simplify the expression in Equation 7 without ruining the variance estimate.

Using Equation 5 and Equation 7, we can approximate the variance with the following expression:

$$\mathbb{V}[t] = \sigma_t^2 \approx \frac{1}{N}\left(\sum_{i=1}^{N}\left(\mathbf{w}\cdot\mathbf{a}^*(\mathbf{x_i})^T\right)^2 - \frac{1}{N}\left(\mathbf{w}\sum_{i=1}^{N}\mathbf{a}^*(\mathbf{x_i})^T\right)^2\right) \tag{8}$$

5 Prioritization

Using the results from Section 3 and 4, i.e., the threat estimation and sensitivity, the two values can now be combined into a task priority value $\pi_k(\mu_t, \sigma_t^2)$ for task k. $\pi_k(\cdot)$ should have the properties that it yields high priority when threat variance is high, since if the variance is high our expected threat is uncertain. Furthermore, if both the expected threat and its variance are low, the priority should be low, since its unit poses no threat and we are pretty certain about that. Does this mean that we will fail to detect the attack of some enemy platoon just because we considered it harmless at some earlier and subsequently didn't bother to look at it again? The answer is "no" since the lack of observations will make the position uncertainty of the unit increase which, eventually, leads to a higher threat variance. High expected threat and low variance should lead to a moderate priority (somewhere in between high and low priority). The reason is that since the threat of the enemy unit is already fairly well established, resources could be devoted to estimating the threat of other enemy units.

Once again, for illustration, we propose a weighted linear combination; π_k is the priority of task k calculated with weights $\mathbf{u} = (u_\mu, u_\sigma)$:

$$\pi_k(\mu_t, \sigma_t^2) = u_\mu\cdot\mu_t + u_\sigma\cdot\sigma_t^2 \tag{9}$$

We propose Algorithm 1 to compute the priority value. The `for`-loop from Line 4 to Line 7 samples for each iteration from $N(\hat{\mathbf{x}}, \hat{\mathbf{P}})$. We simplify the sampling of the three dimensional $\hat{\mathbf{x}}$ by assuming that its three components are independent Gaussian variables. In Line 5 the function `Calculate_distr` that calculates the distribution of probabilities over plans, i.e., $\mathbf{a}^*$, is called. The following two lines of the loop accumulates the contribution of the samples. In Line 8 the approximation of μ_t is completed and in Line 8 the variance is computed. Finally, on Line 11 a function, `Calculate_prio`, is called that combines the expected value and variance of the threat. If $\mathcal{O}(\omega)$ is the time complexity of calculating the probability distribution over plans, then the time complexity of Algorithm 1 is $\mathcal{O}(\omega N)$, where N is the number of samples.

Algorithm 1: Task priority calculation

Input: A position estimate $\hat{x}$, position covariance estimate $\hat{\mathbf{P}}$, the number of samples N

Output: a scalar priority value π

PRIORITY($\hat{\mathbf{x}}$, $\hat{\mathbf{P}}$, N)

    $\mathbf{sum} \leftarrow \mathbf{0}$
    $sq\_sum \leftarrow 0$
    for $i = 1$ **to** N
        $\mathbf{x_i} \leftarrow$ sample from $N(\hat{\mathbf{x}}, \hat{\mathbf{P}})$
        $\mathbf{a_i^*} \leftarrow$ CALCULATE_DISTR($\mathbf{x_i}$)
        $\mathbf{sum} \leftarrow \mathbf{sum} + \mathbf{a_i^*}$
        $sq\_sum \leftarrow sq\_sum + \left(\mathbf{w}\cdot\mathbf{a_i^*}^T\right)^2$
    $exp\_t \leftarrow \mathbf{w}\cdot\mathbf{sum}^T/N$
    $exp\_sqr \leftarrow (exp\_t)^2$
    $var\_t \leftarrow sq\_sum/N - exp\_sqr$
    $\pi \leftarrow$ CALCULATE_PRIO(exp_t, var_t)

If there are more than one friendly platoon that generates tasks we can extend the prioritization function π in Equation 9 with the factor of the relative (strategic) value of the friendly platoon in question. E.g., the value priority of a task generated by friendly platoon i is scaled with a value v_i reflecting i:s significance to the overall mission of the friendly platoons,

$$\pi_k(\mu_t, \sigma_t^2; i) = v_i(u_\mu\cdot\mu_t + u_\sigma\cdot\sigma_t^2). \tag{10}$$

In an application where distributed plutoons (fusion nodes) prioritize their own tasks before submitting them to a sensor management node that performs the *allocation scheme* (i.e., the process that decideds in what order tasks should be served), we can think of the submitted priorities as *local prorities.* For this type of application, let us re-denote the priority function in Equation 9 by $\pi_k^{lcl(i)}$, i.e., the local priority of task k by node i. The sensor management node then, based on the information it has about the mission objectives, generates *global priorities* (or mission priorities), e.g., $\pi_{i.k}^{gbl}(\pi_k^{lcl(i)}; i) = v_i\cdot\pi_k^{lcl(i)}$.

Comments

- Should we use "plan" instead of "activity"?
- Should we use "platoon" and "company"? Check!