Explained Activity Recognition with Computational Assumption-based Argumentation

Xiuyi Fan[1], Siyuan Liu[1], Huiguo Zhang[1], Cyril Leung[2], Chunyan Miao[1]

Abstract. Activity recognition is a key problem in multi-sensor systems. In this work, we introduce *Computational Assumption-based Argumentation*, an argumentation approach that seamlessly combines sensor data processing with high-level inference. Our method gives classification results comparable to machine learning based approaches with reduced training time while also giving explanations.

1 Introduction

We present an argumentation based approach for activity recognition. *Computational Assumption-based Argumentation (CABA)* is an argumentation framework that connects low-level sensor data processing with high-level argumentative reasoning. In the proposed CABA framework, sensor data is processed to form arguments. Together with pre-defined arguments based on domain knowledge representing activities, they jointly construct argumentative inferences such that "winning arguments" represent recognized activities.

CABA frameworks are extensions of the widely recognized Assumption-based Argumentation (ABA) frameworks [4] with added *Computation Units (CUs)*. A CU represents a purposefully designed (numerical) computation that is difficult to represent with plain ABA. CUs are seamlessly built into CABA arguments in ways assumptions are built into ABA arguments. With CUs introduced, the *acceptability* of a CABA argument depends on attack relations and its CUs. Thus, low-level data processing is "packaged" into CUs whereas high-level reasoning as defined by standard argumentation semantics. Upon recognizing activities, explanation for classification is provided. We leverage our previous work on argumentation explanation [1, 2, 3] in the development of CABA explanation.

We test our CABA based activity recognition algorithm in a smart home equipped with: (1) two Grid-Eye inferred sensors[3], (2) two force sensors, (3) one noise sensor, and (4) one electric current detector. We focus on six activities: (1) eat, (2) watch TV, (3) read books, (4) sleep, (5) visit and (6) other. We assume that at any moment, there is one and only one activity taking place.

2 CABA and Explained Activity Recognition

Computation Units (CUs) are core components of CABA. Formally:

Definition 1. A *computation unit (cu)* is a tuple $u = \langle \mathsf{T}, \mathsf{C}, \mathsf{E} \rangle$:
• $\mathsf{T} \subseteq \mathtt{D}_u$: T is the *Data*, and $\mathtt{D}_u$ is the *Domain*;
• $\mathsf{C} : \mathtt{D}_u \mapsto \mathtt{R}_u$: C is the *Computation Function (Computation)*, and $\mathtt{R}_u$ is the *Range*;
• $\mathsf{E} : \mathtt{R}_u \mapsto \{\top, \bot\}$: E is the *Evaluation Function (Evaluation)*.
We say that $u = \langle \mathsf{T}, \mathsf{C}, \mathsf{E} \rangle$ is *successful* iff $\mathsf{E} = \top$ and *well-formed* iff both C and E are total and computable.

In this work, we focus on well-formed CUs. We introduce four CUs: u_{tv}, u_{tb}, u_{bed} and u_{one} representing whether the TV is on, the table is occupied, the bed is occupied and there is a single person, respectively.
1. **TV on:** $u_{tv} = \langle \mathsf{T}_{tv}, \mathsf{C}_{tv}, \mathsf{E}_{tv} \rangle$ in which:
$\mathsf{T}_{tv} \subseteq \{0, 1\}$ is the output from the current sensor;
$\mathsf{C}_{tv}(\mathsf{T}_{tv}) = \mathsf{T}_{tv}$; and
$\mathsf{E}_{tv} = \top$ if $\mathsf{C}_{tv} = 1$ and $\mathsf{E}_{tv} = \bot$ otherwise.
2. **Table occupied:** $u_{tb} = \langle \mathsf{T}_{tb}, \mathsf{C}_{tb}, \mathsf{E}_{tb} \rangle$ in which:
$\mathsf{T}_{tb} \subseteq \mathbb{N}^{8\times 8}$ is the output from GridEye 1;[4]
$\mathsf{C}_{tb}(\mathsf{T}_{tb}) = CL_{tb}(\mathsf{T}_{tb}, \omega_{tb})$ is a classifier such that, with some parameter ω_{tb}, $CL_{tb}(\mathsf{T}_{tb}, \omega_{tb}) \in \{\top, \bot\}$ indicates whether there is any person sitting next to the dining table; and
$\mathsf{E}_{tb} = \mathsf{C}_{tb}$.
3. **Bed occupied:** $u_{bed} = \langle \mathsf{T}_{bed}, \mathsf{C}_{bed}, \mathsf{E}_{bed} \rangle$ in which:
$\mathsf{T}_{bed} \subseteq \mathbb{N}^{8\times 8}$ is the output from GridEye 2;
$\mathsf{C}_{bed}(\mathsf{T}_{bed}) = CL_{bed}(\mathsf{T}_{bed}, \omega_{bed})$ is a classifier such that, with some parameter ω_{bed}, $CL_{bed}(\mathsf{T}_{bed}, \omega_{bed}) \in \{\top, \bot\}$ indicates whether there is any person on the bed; and
$\mathsf{E}_{bed} = \mathsf{C}_{bed}$.
4. **One person in room:** $u_{one} = \langle \mathsf{T}_{one}, \mathsf{C}_{one}, \mathsf{E}_{one} \rangle$ in which:
$\mathsf{T}_{one} \subseteq \mathbb{N}^{132\times 1}$ is the output from all of our sensors;
$\mathsf{C}_{one}(\mathsf{T}_{one}) = CL_{one}(\mathsf{T}_{one}, \omega_{one})$ is a classifier such that, with some parameter ω_{one}, $CL_{one}(\mathsf{T}_{one}, \omega_{one}) \in \{\top, \bot\}$ indicates whether there is a single person in the room; and
$\mathsf{E}_{one} = \mathsf{C}_{one}$.

Definition 2. *Computational Assumption-based Argumentation frameworks* are tuples $\langle \mathcal{U}, \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle$ where
• $\mathcal{U}$ is a set of well-formed CUs;
• $\langle \mathcal{L}, \mathcal{R}, \mathcal{U} \rangle$ is a deductive system, with $\mathcal{L}$ the *language* and $\mathcal{R}$ a set of *rules* of the form $s_0 \leftarrow s_1, \ldots, s_m (m \geq 0, s_i \in \mathcal{L} \cup \mathcal{U}$ for $i > 0$, $s_i \in \mathcal{L}$ for $i = 0)$;
• $\mathcal{A} \subseteq \mathcal{L}$ is a (non-empty) set, whose elements are *assumptions*;
• $\mathcal{C}$ is a total mapping from $\mathcal{A}$ into $2^{\mathcal{L}} - \{\{\}\}$, where each $s \in \mathcal{C}(a)$ is a *contrary* of a, for $a \in \mathcal{A}$.
Given a rule ρ of the form $s_0 \leftarrow s_1, \ldots, s_m$, s_0 is referred as the *head* and $s_1, \ldots, s_m$ as the *body* of ρ.

We use the following CABA framework for activity recognition.
• $\mathcal{U}$ contains the following CUs: u_{tv} u_{tb} u_{bed} u_{one}

---

[1] Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore
[2] Electrical and Computer Engineering, The University of British Columbia, Canada
[3] https://na.industrial.panasonic.com/products/sensors/sensors-automotive-industrial-applications/grid-eye-infrared-array-sensor
[4] The output of a GridEye is an 8-by-8 integer matrix.

• $\mathcal{L}$ contains the following sentences:

watchTV	eat	sleep	visit
notWatch	notEat	notSleep	notVisit
read	other	notRead	noAct
TVon	tableOccupied	bedOccupied	onePerson
TVoff	emptyTable	emptyBed	twoAct

• $\mathcal{R}$ contains the following rules:

notWatchTV ← TVoff	notEat ← emptyTable
notSleep ← emptyBed	notVisit ← onePerson
notRead ← TVon	notRead ← tableOccupied
notRead ← bedOccupied	other ← noAct
twoAct ← watchTV, eat	twoAct ← watchTV, sleep
twoAct ← watchTV, visit	twoAct ← watchTV, read
twoAct ← eat, sleep	twoAct ← eat, visit
twoAct ← eat, read	twoAct ← sleep, visit
twoAct ← sleep, read	twoAct ← visit, read
TVon ← u_{tv}	tableOccupied ← u_{tb}
bedOccupied ← u_{bed}	onePerson ← u_{one}

• $\mathcal{A}$ contains the following assumptions:

watchTV	eat	sleep	visit	read
TVoff	emptyTable	emptyBed	noAct	

• $\mathcal{C}$ are:

$\mathcal{C}$(noAct) = {watchTV, eat, sleep, read, visit}
$\mathcal{C}$(watchTV) = {notWatchTV, twoAct} $\mathcal{C}$(eat) = {notEat, twoAct}
$\mathcal{C}$(sleep) = {notSleep, twoAct} $\mathcal{C}$(read) = {notRead, twoAct}
$\mathcal{C}$(visit) = {notVisit, twoAct} $\mathcal{C}$(TVoff) = {TVon}
$\mathcal{C}$(emptyTable) = {tableOccupied} $\mathcal{C}$(emptyBed) = {bedOccupied}

We define CABA arguments and attacks as follows.

Definition 3. A CABA argument for (claim) $s \in \mathcal{L}$ *supported by* $\Delta \subseteq \mathcal{A}$ *based on* $U \subseteq \mathcal{U}$ (denoted $[\Delta, U] \vdash s$) is a finite tree with nodes labeled by sentences in $\mathcal{L}$, CUs in U or by $\tau \notin \mathcal{L} \cup \mathcal{U}$, the root labeled by s, leaves labeled by either τ, assumptions in Δ, or CUs in U, and non-leaves labelled by s' with, as children, sentences in the body of some rule with head s'.

A CABA argument $A = [\Delta, U] \vdash s$ is *applicable* iff for all CUs $u = \langle \mathsf{T}_u, \mathsf{C}_u, \mathsf{E}_u \rangle \in U$, $\mathsf{E}_u = \top$. For a CABA argument $A = [\Delta, U] \vdash s$, if $U = \{\}$, then A is abbreviated to $\Delta \vdash s$. Given a CABA framework *F*, an argument *is in F* iff all its rules, assumptions and CUs are in *F*. $\mathtt{A}^F$ denotes the set of all arguments in *F*.

Definition 4. Given a CABA framework *F*, an argument $[A_1, U_1] \vdash s_1$ (in *F*) *attacks* an argument $[A_2, U_2] \vdash s_2$ (in *F*) iff s_1 is a contrary of some assumption in A_2. $\mathtt{R}^F$ denotes the set of all attacks in *F*.

We let admissibility apply in CABA with additional conditions: (1) a set of arguments is admissible only if they are applicable and (2) an admissible set of arguments only needs to counter-attack all attacks from applicable attackers. We formalize explanations for non-admissible CABA arguments as follows.

Definition 5. Given a CABA framework *F* with arguments $\mathtt{A}^F$ and attacks $\mathtt{R}^F$, and CABA argument $A \in \mathtt{A}^F$ such that A is not admissible in *F*, then, if there exists some $As \subseteq \mathtt{A}^F$, such that: (1) A is admissible in $\langle \mathtt{A}^F, \mathtt{R}^F \rangle \setminus As$, and (2) there is no $As' \subset As$ such that A is admissible in $\langle \mathtt{A}^F, \mathtt{R}^F \rangle \setminus As'$, then As is an *explanation* of A. Otherwise, $\{A\}$ is the *explanation* of A.

Suppose that for some sensor data, both u_{tb} and u_{one} are successful whereas u_{tv} and u_{bed} are not successful. We see that {watchTV} ⊢ watchTV is not admissible. However, if arguments
$A = \{\text{TVoff}\} \vdash \text{notWatchTV}$, $B = [\{\}, u_{tb}] \vdash \text{TableOccupied}$ are removed, then {watchTV} ⊢ watchTV becomes admissible. Hence, $\{A, B\}$ is an explanation for {watchTV} ⊢ watchTV. We can interpret this as:

An explanation for "not watching TV" is that the TV is off and there is a person using the dining table.

To evaluate CABA based activity recognition, we have had four individuals performing the six activities. Each person performs the six activities in two runs. Data collected from six runs from three testing subjects are used for training with the remaining two runs from the fourth subject for testing. Overall, there are 7781 instances of training samples and 1437 instances of testing samples.

With the CABA framework presented earlier, to perform activity recognition we use four CUs, u_{tv}, u_{tb}, u_{bed} and u_{one}. We let u_{tv} use the value directly from the current sensor. To construct u_{tb} and u_{one}, we choose perceptron classifiers for their simplicity. Specifically, for u_{tb}, we construct a 64-node perceptron model with a single output node; for u_{one}, we construct a 132-node perceptron model with a single output node. To construct u_{bed}, we use binary thresholding based on GridEye 2's output to test if the bed is occupied.

We compare our CABA-based classifier with Naive Bayes, Decision Tree and Neural Networks using precision, recall and training time. The results are summarized in Table 1. These results illustrate the effectiveness of introducing domain knowledge in argumentation form in solving activity recognition problems.

	Precision	Recall	Training Time (s)
Naive Bayes	0.558	0.578	2.89
Decision Tree	0.576	0.397	0.46
NN 2-Hidden Layers	0.801	0.797	549.32
Deep Neural Network	0.816	0.812	1865.34
CABA-Classifier	0.821	0.831	53.55

Table 1. Performance comparison between Naive Bayes, Decision Tree, Neural Networks 2 and 8 hidden layers and the proposed CABA-classifier.

3 Conclusion

In this work, we presented CABA to seamlessly connect low-level data processing with high-level inference based reasoning. We used CABA to solve an activity recognition problem, with promising results comparable to traditional machine learning algorithms. The advantage of CABA is twofold. Firstly, used as a channel for injecting domain knowledge into problem solving, CABA significantly reduces the training time required for model construction. Secondly, the argumentative structure of CABA provides the basis for generating explanations for the modeled computation.

Acknowledgments

This research was supported by the National Research Foundation Singapore under its Interactive Digital Media (IDM) Strategic Research Programme.

REFERENCES

[1] X. Fan and F. Toni, 'On computing explanation in abstract argumentation', in *Proc. ECAI*, (2014).
[2] X. Fan and F. Toni, 'On computing explanations for non-acceptable arguments', in *Proc. TAFA*, (2015).
[3] X. Fan and F. Toni, 'On computing explanations in argumentation', in *Proc. of AAAI*, (2015).
[4] F. Toni, 'A tutorial on assumption-based argumentation', *Argument & Computation, Special Issue: Tutorials on Structured Argumentation*, **5**(1), 89–117, (2014).