1-1-2019

Active Perception by Interaction with Other Agents in a Predictive Coding Framework: Application to Internet of Things Environment

Masoumeh Heidari Kapourchali

ACTIVE PERCEPTION BY INTERACTION WITH OTHER AGENTS IN A PREDICTIVE CODING FRAMEWORK: APPLICATIONS TO INTERNET OF THINGS ENVIRONMENT

by

Masoumeh Heidari Kapourchali

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Major: Engineering

The University of Memphis
December 2019

To my parents and my only brother, Mohammad

Acknowledgments

I would like to give my utmost thanks and deepest appreciation to my advisor Dr. Bonny Banerjee to whom I am and will always remain indebted. This dissertation would not have been possible without his constant help and support. I feel privileged to work under his supervision for five years of my graduate studies. His substantial and thorough approach, dedication, and attitude toward research are guides for my future academic activities. Thank you for being a wonderful mentor Dr. Banerjee.

I would like to extend my gratitude to my dissertation committee members, Dr. Bowman, Dr. Balasubramanian, Dr. Jacobs and Dr. Venugopal for their kind help and support, as well as valuable comments and feedback. My mere expression of thanks does not suffice.

I also thank Institute for Intelligent Systems (IIS) for supporting our research, providing me the opportunity of attending academic conferences through which I have learned a lot, as well as showing me the benefits of performing interdisciplinary research. And of course, I appreciate Computational Intelligence Lab, Department of Electrical and Computer Engineering, and University of Memphis.

My special thanks goes to my amazing family. To my parents Pari Sohrabi and Hassan Heidari, not only for their unconditional love and continuous support but also for teaching me the importance of following my dreams. For always reminding me all my dreams are within my grasp and encouraging me to move forward. To my brother, Dr. Mohammad Heidari for being my source of inspiration and a trusted role model. For teaching me the power of hard work and informing me I have merely completed a stage but I am nowhere near the end of this journey.

ABSTRACT

Predicting the state of an agent's partially-observable environment is a problem of interest in many domains. Typically in the real world, the environment consists of multiple agents, not necessarily working towards a common goal. Though the goal and sensory observation for each agent is unique, one agent might have acquired some knowledge that may benefit the other. In essence, the knowledge base regarding the environment is distributed among the agents. An agent can sample this distributed knowledge base by communicating with other agents. Since an agent is not storing the entire knowledge base, its model can be small and its inference can be efficient and fault-tolerant. However, the agent needs to learn – when, with whom and what – to communicate (in general interact) under different situations.

This dissertation presents an agent model that actively and selectively communicates with other agents to predict the state of its environment efficiently. Communication is a challenge when the internal models of other agents is unknown and unobservable. The proposed agent learns communication policies as mappings from its belief state to when, with whom and what to communicate. The policies are learned using predictive coding in an online manner, without any reinforcement. The proposed agent model is evaluated on widely-studied applications, such as human activity recognition from multimodal, multi-source and heterogeneous sensor data, and transferring knowledge across sensor networks. In the applications, either each sensor or each sensor network is assumed to be monitored by an agent. The recognition accuracy on benchmark datasets is comparable to the state-of-the-art, even though our model has significantly fewer parameters and infers the state in a localized manner. The learned policy reduces number of communications. The agent is tolerant to communication failures and can recognize the reliability of each agent from its communication messages. To the best of our knowledge, this is the first work on learning communication policies by an agent for predicting the state of its environment.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Predicting the state of an agent's partially-observable environment is a problem of interest in many domains, such as medicine, surveillance and economy. Typically in the real world, the environment consists of multiple agents, not necessarily working towards a common goal. Though the goal and sensory observation for each agent is unique, one agent might have acquired some knowledge that may benefit the other. In essence, the knowledge base regarding the environment is distributed among the agents. An agent can sample this distributed knowledge base by communicating with other agents. Since an agent is not storing the entire knowledge base, its model can be small and its inference can be efficient and fault-tolerant. However, the agent needs to learn when, with whom and what to communicate under different situations.

This dissertation investigates how an agent can optimally communicate with other agents for predicting the state of its environment. We model communication as an action that facilitates active perception [BAT18] whereby an agent actively and selectively samples (or *communicates with*) other agents. Communication makes causal knowledge acquisition efficient by allowing to: (1) share causal knowledge regarding the same event even though the observations are from different sensors in space, time or modality, and (2) acquire high-level causal knowledge directly from another agent instead of from the low-level sensory environment. Hence, communication by an agent is inevitable for predicting its environmental state efficiently. Communication is a challenge when the internal models of other agents is unknown and unobservable. The proposed agent learns communication policies as mappings from its belief state to when, with whom and what to communicate. The policies are learned using predictive coding in an online manner, without any reinforcement.

The proposed agent model is evaluated on widely-studied applications, such as human activity recognition from multimodal, multisource and heterogeneous sensor data, detecting Freezing of the Gait (Fog) in Parkinson's disease patients, and transferring knowledge of activity recognition in Internet of Things Environment. In the applications, either each sensor or each sensor network is assumed to be monitored by an agent. The recognition accuracy on benchmark datasets is comparable to the state-of-the-art, even though our model has significantly fewer parameters and infers the state in a localized manner. The learned policy reduces number of communications. The agent is tolerant to communication failures and can recognize the reliability of each agent from its communication messages.

1.1 Related work

Estimating the state of the environment from sensor data (a.k.a. *state estimation* or *filtering*) is a core component in many applications. A taxonomy tree of state estimation methods, including the proposed one, is shown below.

```
State Estimation
└─Passive Estimation
  ├─Single Sensor
  │ ├─Single Estimator [GSS93, WVDM00, GMH13]
  │ └─Multiple Estimators [JM08]
  └─Multiple Sensors
    ├─Single (Central) Estimator [EK99]
    └─Multiple Estimators [OS05, CS10]
      ├─Communicate with a Central Center
      └─Communicate Locally
        └─Predefined Protocol
```

- Active Perception
  - Single Sensor
    - Single Decision Maker [JG18, YLP+15, YLW16, EGZ09, DB02, GHRA17, ASLN+14, Vas18, BM09]
  - Multiple Sensors
    - Single (Central) Decision Maker [ZSR02, JPC07, VR16, BDN+16, KHG11, Spa08, QAT+15, ZJ05, ZJ06, KKHI05, LJRW16, SWO15, SWOS18]
    - Multiple Decision Makers
      - Communicate with a Central Center [SF17]
      - Communicate Locally
        - Predefined Protocol [WYL+18, BCP+18, WSH03, YYL03, ZYO+15, KB18a, KB19]
        - Learned Policy
          - Globally Learned
          - Locally Learned [proposed model]

Distributed control and filtering are widely studied in networked systems where a large number geographically dispersed, cooperative and life-critical systems with a large number of sensors and actuators work together to facilitate real-time monitoring and closed-loop control [DHWG19]. Distributed filtering algorithms using consensus [OS05] and diffusion [CS10], and decentralized control are widely used in multiagent systems [NM14]. Periodic all-to-all communication for data exchange between components causes large communication burdens [NGC19]. Developing event-triggered communication protocols have attracted significant attention [ZWZ17] in recent years to avoid unnecessary data transmission. In such systems, each agent broadcasts its state information to its neighbors *when* an internal error signal exceeds a state-dependent threshold [MT08]. In our model, the agents are not forced to reach a consensus as they can have distinct goals. Learning communication policies allows an agent to autonomically and independently decide when, with whom and what to communicate. This is particularly beneficial where a neighbor might not be geographically close to the agent, and neighbors change with space, time or the agent's goals.

1.2 Overview of contributions

- We propose a well-defined mechanism that allows autonomous, self-interested artificial agents to decide *when*, with *whom*, and *what* to communicate.

- We propose an agent model that generalizes predictive coding [FDK09] and active perception [BAT18] for an environment containing other agents where the knowledge required for the agent's environmental state estimation is distributed among multiple agents. To the best of our knowledge, this is the first work on learning communication policies by an agent for predicting the state of its environment. Predictive coding is a closed-loop approach with no need for supervision (labels) or reinforcement.

- Unlike existing models that learn what or with whom to communicate (such as [Hos17, DGR+18, HYZW18]), our model learns and executes communication policies in a localized manner (i.e. it communicates neither with a central/global controller nor with all the agents all the time).

- The communication policies learned by our agent adapt to changes in other agents' behavior. Modeling changing behaviors of other agents when their internal model is partially observable and unknown is an open problem in artificial intelligence (AI) [AS18]. Our agent learns a model of each communicating agent as a mapping from that agent's communication messages to its belief.

- Human activity recognition from multimodal, multisource and heterogeneous sensor data is used as a testbed to evaluate the proposed model. Experimental results on benchmark datasets show that the environmental state prediction accuracy of our model is comparable to the state-of-the-art even though it uses significantly fewer parameters. The model is efficient in terms of number of communications, and tolerant to communication failures.

- Our model has ability of learning interpretable patterns for activities of daily living in a smart home setting. The knowledge of activity recognition can be transferred to other agents for whom limited or no training data is available.

1.3 Outline

This dissertation will proceed as follows:

Chapter 2: Predictive coding. chapter 2 will cover some background materials on artificial intelligent agents, the widely studied architectures and agent types, as well as predictive coding. Then the architecture for our predictive coding agent is introduced. The agent's mechanism for perception, action selection and learning using the predictive coding agent is briefly discussed and agent's model is compared with traditional agents in artificial intelligence literature.

Chapter 3: Communication with other agents. Using a controlled experiment, we discuss benefits of communicating with other agents for estimating state of a shared environment. We will show that sensory limitations may lead to incorrect or delayed causal inferences giving rise to conflicts in the mind of a predictive coding agent, and communication helps to resolve such conflicts and overcome the limitations.

Chapter 4: When to communicate. This chapter covers advantages of opportunistic communication by a predictive coding agent for estimating the state of its environment. The model is applied to action recognition from multimodal datasets as well as freezing of gait (FoG) recognition from wearable acceleration sensors in Parkinson's disease (PD) patients. Communication overhead is minimized using a low rank approximation on previous successful communications. The model can also deal with missing values and sensor failures.

Chapter 5: With whom to communicate. In this chapter, we will show how our predictive agent can learn a communication policy as a mapping from its belief state to with whom to communicate. The learned policy reduces number of communications. The agent is tolerant to communication failures and can recognize the reliability of each agent from its communication messages. The policy adapts to the changes in other agents' behaviors.

Chapter 6: What to communicate. This chapter shows how our predictive coding agents can transfer their knowledge by active and selective communication. Recognizing daily activities in houses with different layouts and devices has been selected as a testbed. The agents Communicate most informative messages and learn interpretable patterns of individuals' behaviors which is important for healthcare applications.

1.4 First published appearances

The results of Chapter 3 appeared in [KB18a]. The unsupervised feature learning algorithm of Chapters 4 and 5 first appeared in [KB18b]. Chapter 4 covers materials for opportunistic communication which has been published in [KB19]. Part of theoretical and experimental results of chapter 5 has been published in [HKB20].

Chapter 2

Predictive coding

The problem of inferring the causes from sensations is ill-posed [KFF07] as different causes can generate the same sensation. Predictive coding [Fri10] is a brain-inspired framework for solving this problem by minimizing variational free energy.

Predictive coding is a leading theory of how the brain performs probabilistic inference [Spr17]. To correctly interpret sensory data, the brain is faced with solving an inverse problem: the causes need to be inferred from the perceived outcomes [RB99]. This problem is ill-posed as different causes can generate the same sensation. Predictive coding suggests that the brain is equipped with an internal model of the world which encodes possible causes of sensory inputs as parameters of a generative model. New sensory inputs are then represented in terms of these known causes. There are multiple versions of predictive coding [Spr17]. Unfortunately, the advantages of this framework in modeling AI agents have not been investigated until recent years as it has been published over different stages of evolution using varying notations so the mathematics remained non-trivial [Bog17, BKMS17, HWZ+18].

2.1 Traditional agent in artificial intelligence

An agent is anything that can perceive its environment through sensors and act upon that environment through actuators [RN16]. Four basic kinds of agent programs are [RN16]: 1) simple reflex agents in which the agents' program contains a set of rules, 2) model-based reflex agents where the agent keeps track of the current state of the environment but chooses an action in the same way as the simple reflex agent, 3) goal-based agents where the agents keeps track of a set of goals it is trying to achieve, and hence chooses actions that lead to the achievement of the goals, and 4) utility-based agents which provide a performance measure to allow a comparison of different world states. There are different approaches for optimizing the utility. Markov decision processes (MDPs) are widely used to provide a *closed-loop non-myopic* solution for agents' optimal decision making problem [RN16]. Partially observable MDPs (POMDPs) is an extension of MDP when the states are partially observable. A POMDP can be converted to a MDP using beliefs about the current state. The belief can be recursively computed from the observations and actions using Bayes rule.

2.2 Predictive coding agent

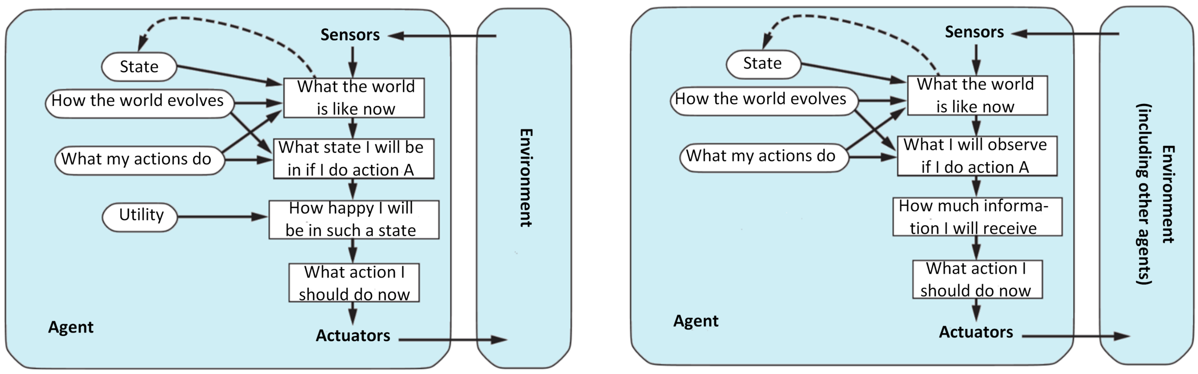
Predictive coding is a brain-inspired framework for solving the problem of inferring the causes from sensations [RB99]. Inspired by linearly solvable MDPs [Tod07] and path integral control frameworks [KGO12], a version of predictive coding, active inference, proposes an alternative approach for modeling an agent which is efficient and does not require a reward function to compute optimal policy [Fri10]. By modeling action as inference and maximizing marginal likelihood of observations under a generative model, the optimal policy can be computed as a Kullback-Leibler (KL)-divergence minimization problem. The mathematical proof is provided in [FDK09] to show that these *policies are equivalent to the ones computed through Bellman optimality equation. Hence predictive coding is a generalization of optimal control or POMDPs.*

An agent in active inference framework is defined as the tuple $\langle \Psi, A, \vartheta, G, Q, R, \Phi \rangle$ where Ψ is a set of states, A is a set of actions. ϑ is a set of real valued parameters. G and Q are generative and recognition densities, respectively. R is sampling probability and Φ is a set of sensory states [FSM12]. The agent's objective is to minimize *Variational free energy* (VFE) which is a measure of salience based on the divergence between the recognition density $Q(\psi)$ and generative density $p(\varphi, \psi)$ [Fri10]: $F = -\langle \ln p(\varphi, \psi) \rangle_Q + \langle \ln Q(\psi) \rangle_Q$ where $\langle . \rangle_Q$ denotes the expectation under density Q.

In this work, each agent is modeled using active inference framework and is called *predictive coding agent*.

2.3 Comparison between a predictive coding and a traditional AI agent

Figure 2.1 shows the difference between a standard utility-based agent and our proposed agent. The predictive coding agent does not use reinforcement or utility to optimize action. Action is optimized by minimizing VFE which is a function of observation and recognition density (internal representation of state) or its sufficient statistics. Cost function is replaced by priors over states and transitions so that the agent chooses an action which reveals the most informative observation. Similar to POMDPs, the states and observations are distinct; however, transition function is a mapping between actions and direct sensory consequences (sampling probability), so probabilistic observations are conditioned upon actions. Action simply serves to realize posterior beliefs about state transition. In contrast, utility is conditioned upon states in POMDPs.



(a) Utility-based agent (adopted from Chapter 2 of [RN16]).

(b) Proposed predictive coding agent.

Fig. 2.1: Our proposed agent in contrast to a standard utility-based agent often modeled using POMDPs or reinforcement learning.

Chapter 3

Communication with other agents

3.1 Introduction

In this chapter, agents are embodied multimodal entities and situated in a shared environment. They have different visibility of the environment due to unique sensory and generative models. Similar to biological entities, none of the agents can completely observe the reality. Each agent's version of the world is a function of its sensory system and internal model. In the context of decision-making about current state of environment, each agent's limitations can be overcome to some extent through communication with the other agents. Communication extends an agent's perceptual field and allows efficient causal knowledge acquisition by sampling other agents' internal causal models. We show that communication between agents helps each of them reach a shared decision in a way that cannot be reached by brain processes in a single agent. Using a simulated environment, we show that sensory limitations may lead to incorrect or delayed causal inferences giving rise to conflicts in the mind of a predictive coding agent, and communication helps to resolve such conflicts and overcome the limitations.

In this chapter, we propose a computational model of state estimation by multiple predictive coding agents through mutual communication about a common subject. In Figure 3.1, this problem is illustrated using two agents, A_1 and A_2, communicating regarding a third entity, A_3. The sensory/generative system of each agent is unique. To make the problem interesting, each agent is assumed to be multimodal, receiving sensory observa-

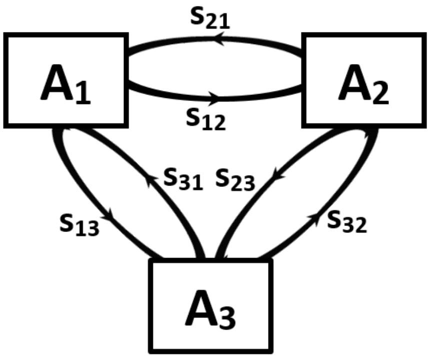


Fig. 3.1: Distributed decision-making through communication between two agents, A_1 and A_2, regarding the state of a third entity, A_3. s_{ij} denotes a signal passed from entity A_i to entity A_j which could be a communicative message or a sample of an observable variable related to the state of an entity.

tion in two modalities: one, directly from the common subject, and the other from the other agent(s) due to communication about the subject. The former modality is unique for each agent while the latter modality (for receiving communicative inputs) is common across all agents. At any instant of time, the goal is for all communicating agents to reach a decision on the state of the common subject more accurately and quickly than each of them could have by itself. This goal stems from social cognition research [DPDJ12] where communication is construed as dynamic interaction among multiple individuals which helps reach a shared decision in a way that could not be reached by brain processes in a single individual. Communication in our model is at the level of agents' beliefs and is not limited to low-level brain/spinal signals.

3.2 Definitions

The terms and concepts relevant to this chapter are discussed in this section.

Definition 2. **(Hermeneutic circle)** The hermeneutic circle is used as a model of explaining communication. It refers to the problem of circularity of understanding [CM11] where understanding the first agent presupposes understanding the second agent, which in turn presupposes understanding the first agent [FF15].

Definition 3. (Variational free energy) Variational free energy is a measure of salience based on the divergence between the recognition $q(x)$ and generative density $p(\varphi, x)$ [FDK09]: $F = - < \ln p(\varphi, x) >_q + < \ln q(x) >_q$.

Definition 4. (Recognition density) Recognition density $q(x)$, is a probabilistic representation of causes which is encoded by internal states μ. Assuming it as a Gaussian density, it is also called Laplace approximation [Fri10]: $q(x) = \mathcal{N}(x; \mu, \zeta) = \frac{1}{\sqrt{2\pi\zeta}} exp\{-(x - \mu)^2/2\zeta\}$

Definition 5. (Generative density) Generative density $p(\varphi, x)$ is a joint probability density relating environmental states and sensory data. It is usually specified in the form of a prior $p(x)$ and a likelihood $p(\varphi|x)$ [BKMS17]. Variables used in this chapter are listed in Table 5.1.

Table 3.1: Symbols and notations for Chapter 3.

Variable	Description
φ	Sensory data
μ	Belief (expectation of cause)
ϵ_φ	Sensory prediction error
ϵ_p	Prior prediction error
σ_φ	Variance of generative density
σ_p	Variance of prior density
x	Environmental variables
x_p	Mean of prior density
a	Action
f_s	Sampling frequency

3.3 Models and methods

The interaction of a group of embodied agents is modeled to infer the states of the environment in which all the agents are situated. The environment is partially-observable to each agent due to their sensory limitations.

3.3.1 An agent and its environment

In our framework, each agent has a unique internal model and shares the environment with other agents. In addition to perception of the shared environment, each predictive coding agent is required to have a model of other agents as part of its internal model to anticipate their future actions. Each agent can act on its environment and change its state. Therefore, even though the agents are independent entities, their actions and perceptions are not entirely independent. Figure 3.2 shows the diagram of an environment shared by two predictive coding agents, each with a generative internal model. The two agents have unique sensors and effectors, and can act on and perceive from the shared environment. The environmental states cannot be observed directly and have to be inferred from sensory observations. Similarly, the state inferred by the other agent is also unobservable and may be estimated from the sensory observations of that agent's behaviors.

As a running example throughout this chapter, consider two agents trying to infer the state of their common environment. One agent is equipped with a sensor that senses the light intensity (a.k.a. *light-agent*) while the other agent is equipped with a sensor that senses the temperature (a.k.a. *heat-agent*). At any time instant, the environment can be in one of three states: *noEvent*, $firework$ or $fire$. Each agent's goal is to infer the state of the environment at all times. The environment is modeled as:

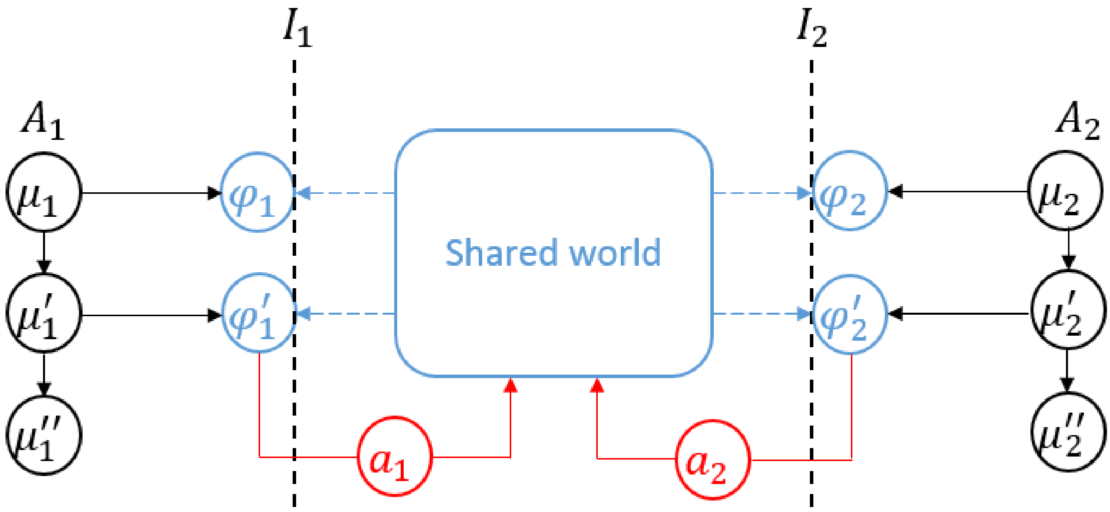


Fig. 3.2: A schematic representation of two predictive coding agents with unique internal (generative) models in a shared environment (modified from [BKMS17]). Interactions between generalized internal states (black) and sensory data (blue) are shown. The agents' actions on the world are represented by a_i (red). I_1, I_2 are the interfaces for agents A_1, A_2 respectively. Everything to the right of I_1, including A_2 and the shared environment, is considered as the external environment for A_1. [Best viewed in color.]

$$f(x) = \begin{cases} noEvent, & \text{if } x < 0.1 \\ firework, & \text{if } 0.1 \leq x < 1 \\ fire, & \text{if } x \geq 1 \end{cases} \tag{3.1}$$

where x denotes the state of the environment. Each agent is also equipped with an actuator using which it can sample its environmental signal, such as light intensity or temperature, at a frequency of its choice within a range.

The generative density of agent A_i is given by:

$$p(\varphi_i|x) = \mathcal{N}(\varphi_i; g_i(x), \sigma_{\varphi_i}) \tag{3.2}$$

where $\mathcal{N}$ denotes the normal density with mean $g_i(x)$ and variance σ_{φ_i}, since the observations are noisy. The generative model g is unique for each agent; it is a mapping from the causes to the observations where the observations are function of the sensors or body of the agent. Let A_1 and A_2 be the light-agent and heat-agent respectively. Then g_1 is defined for A_1 as:

$$g_1(x) = \begin{cases} xt^{\alpha_1 - 1}(1-t)^{\beta_1 - 1}, & \text{if } x < 0.1 \\ xt^{\alpha_2 - 1}(1-t)^{\beta_2 - 1}, & \text{if } 0.1 \leq x < 1 \\ xt^{\alpha_3 - 1}(1-t)^{\beta_3 - 1}, & \text{if } x \geq 1 \end{cases} \tag{3.3}$$

where α_k, β_k ($k = 1, 2, 3$) are predefined parameters and t denotes time. Also, g_2 is defined for A_2 following the convection equation, as:

$$g_2(x) = \begin{cases} xh(T_{hot_1} - T_{cold})Bt, & \text{if } x < 0.1 \\ xh(T_{hot_2} - T_{cold})Bt, & \text{if } 0.1 \leq x < 1 \\ xh(T_{hot_3} - T_{cold})Bt, & \text{if } x \geq 1 \end{cases} \tag{3.4}$$

where T denotes temperature in Kelvin, B is the area of exposure, and h is a constant. T_{hot} varies with situations such that a change in temperature due to $fire$ is different from that due to $firework$. The agents are initialized with a prior regarding the environmental states which is assumed to be a normal distribution $\mathcal{N}(\mu; v_{p_i}, \sigma_{p_i})$ with mean v_{p_i} and variance σ_{p_i} for agent A_i. It is assumed that the frequency of sampling, f_{s_i}, by A_i of its environment is proportional to the change in its observation:

$$f_{s_i} = \frac{d\varphi_i(t)}{dt} \tag{3.5}$$

An agent samples the environment using its body which constitutes a behavior that is observable to other agents.

Each agent can independently infer the environmental states by minimizing the free energy, given by:

$$F = \int -q(x) \ln p(x, \varphi) dx + \int q(x) \ln q(x) dx \tag{3.6}$$

where the first term is the average energy and the second therm is negative of entropy associated with the recognition density [BKMS17]. Assuming $q(x)$ to be a sharply peaked Gaussian density function (i.e. the Gaussian bell shape is squeezed towards a delta function), the most likely value of the environmental state is estimated iteratively using Bayesian approximation as follows:

$$\frac{\partial F}{\partial \mu} = \dot{\mu} = \epsilon_\varphi g'(\mu) - \epsilon_p \tag{3.7}$$

where ϵ_φ and ϵ_p are updated as follows:

$$\dot{\epsilon}_p = \mu - x_p - \sigma_p \epsilon_p \tag{3.8}$$

$$\dot{\epsilon}_\varphi = \varphi - g(\mu) - \sigma_\varphi \epsilon_\varphi \tag{3.9}$$

and the prediction errors are $\epsilon_\varphi = (\varphi - g(\mu))/\sigma_\varphi$ and $\epsilon_p = (\mu - x_p)/\sigma_p$. For a detailed derivation of Equation 4.8 from Equation 5.4, refer to [Bog15]. Note that, μ is the belief of an agent from its observation of the environment without being influenced by any agent through communication.

3.3.2 Reading others' minds from their behaviors

In the real world and also in our simulated environment, agents have different sets of knowledge due to differences in sensory systems/body and prior experience. Communication with other agents helps to sample from their knowledge. However, an agent may be so biased towards its own beliefs that it fails to detect its need for communication. In the context of predictive coding, it means that the agent fails to register a prediction error in which case there is no way to improve its perception.

Friston and Frith [FF15] observe that there is no way to verify whether an agent's interpreted cause of another's behavior corresponds to the latter's actual cause or not. The best the agent can do is to invent a coherent story that minimizes all conflicts in its mind. The ability to interpret an agent requires a model of that agent to be learned by observing its behaviors. In addition to predicting the environment, a predictive coding agent should be able to predict the other agents' behaviors. The observable behaviors of our light-agent and heat-agent are their sampling frequencies which are assumed to be noisy. The model of agent A_i in the mind of agent A_j is of the form: $\mu_{ij} = H(f_{s_j}; \theta_H)$ where H is a mapping from f_{s_j} to μ_{ij} given the set of parameters θ_H, and μ_{ij} is A_i's belief based on A_j's behavior which is different from that due to its observation of the shared environment.

3.3.3 Multiple agents and their communication

In order to extend our discussion to multiple agents, the light-agent and heat-agent will be equipped with a sensor that can sense the frequency of sampling of the environment by the other agents. Each agent has two effectors: one for sampling the environment and the other for sending communicative messages to other agents. Thus, each agent receives observations regarding the shared environment from two sources: one directly from the environment via their light/temperature sensors and the other from the communicating agent. A conflict arises in the mind of an agent whenever the inferred causes from these two sources are not in agreement. Such conflicts have to be resolved by further sampling of the environment and communication. There are many approaches in the literature for conflict resolution [ADWW98, Ole99, MD00, SKH14]. We use belief revision based on trust. Trust is measured by an agent's level of confidence regarding its belief. Communication is a language that both agents ought to understand; that is, they are required to have the ability to encode and decode the communicative messages. In our running example, we assume a message to be a function of the other agent's belief (μ_j), written as $msg(\mu_j, \theta_{comm})$, where θ_{comm} is a set of parameters of the model and can be learned from data. After receiving a message from A_j, A_i's belief is revised as follows:

$$\hat{\mu_i} = \underset{x}{\operatorname{argmax}}\, p(x|\varphi_i, msg(\mu_j, \theta_{comm})) \tag{3.10}$$

Assuming the noise components to be independent and using Bayes rule, we get [DP04]:

$$\begin{aligned} p(x|\varphi_i, msg) \propto p(\varphi_i, msg|x) &\propto p(\varphi_i|x)p(msg|x) \\ &\propto p(x|\varphi_i)p(x|msg) \end{aligned}$$

where $p(x|\varphi_i)$ and $p(x|msg)$ are Gaussian probability densities. The bimodal estimate can be a linear combination of the unimodal estimates. For N agents where all agents send messages to A_i except itself, the bimodal estimate is:

$$\hat{\mu_i} = \left(\frac{\mu_i}{\sigma_{p_i}} + \sum_{\substack{n=1 \\ n \neq i}}^{N} \frac{msg_n}{\sigma_{p_n}} \right) \Bigg/ \sum_{m=1}^{N} \frac{1}{\sigma_{p_m}} \tag{3.11}$$

Here $\hat{\mu_i}$ is the belief of A_i after communicating with other agents and weighing their messages. Inverse of σ_{p_j} is a measure of A_i's trust on A_j's message. If all weights are equal, i.e. $\sigma_{p_i} = \sigma_{p_j} \; \forall i, j, \; i \neq j$, the belief of all agents will converge to the same value which will render all agents except one redundant. Learning a unique model of other agents by each agent allows the entire multiagent system to store more knowledge about the shared environment and allows each agent to resolve conflicts with other agents amicably. By sampling from other agents' internal models through communication, each agent acquires causal knowledge more efficiently than by observing the environment as the environment can only present correlations but an agent can share its causal knowledge.

Inverse of variance is a measure of precision in predictive coding [FDK09]. An agent may not have an accurate model of trust from the beginning. To improve the model, the precision is updated along with minimization of free energy. The update rules for parameters of prior density with each observation are as follows [Bog15]:

$$\frac{\partial F}{\partial x_p} = \dot{x}_p = \frac{\mu - x_p}{\sigma_p} = \epsilon_p \tag{3.12}$$

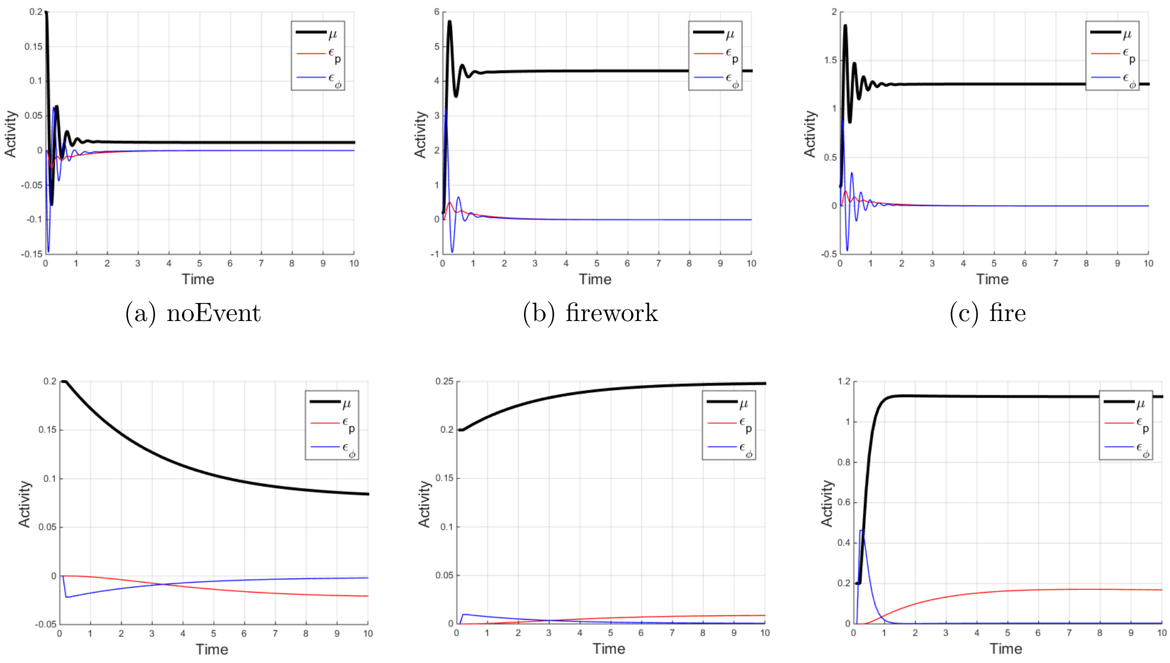
$$\frac{\partial F}{\partial \sigma_p} = \dot{\sigma}_p = \frac{1}{2}(\frac{(\mu - x_p)^2}{\sigma_p^2} - \frac{1}{\sigma_p}) = \frac{1}{2}(\epsilon_p^2 - \sigma_p^{-1}) \tag{3.13}$$

x_p and σ_p converge to mean and standard deviation respectively of an agent's prior density.

3.4 Experimental results

This section discusses the experimental results from applying the proposed distributed decision-making model on the simulated environment for different scenarios consisting of the three events: *noEvent*, $fire$ and $firework$. In particular, we are interested in understanding how the light-agent and heat-agent infer the environmental states, independently and after mutual communication.

Figure 3.3 shows the inference by each agent independently for three observation points: $x = 0$, $x = 0.25$ and $x = 1.25$, representing $noEvent$, $firework$ and $fire$ respectively. The plots show how each agent's belief converges to a particular value of x. $Time$ in these plots refers to the duration of time an agent requires to analyze its observation and for the responses (activities) to settle down. The agent finds the most likely value of x by minimizing the free energy. Two prediction errors are involved in the simulation: ϵ_φ is the difference between observation and its expectation if $x = \mu$, and ϵ_p is the difference between the belief and the prior expectation.



(a) noEvent (b) firework (c) fire

(d) noEvent (e) firework (f) fire

Fig. 3.3: Inference of two agents independently, for a sample of each situation. (Top) Light-agent's inference. (Bottom) Heat-agent's inference. For $firework$, the light-agent (b) converged to $\mu = 4.1$ which is in the range of $fire$. That is, the light-agent made an error in predicting $firework$.

There is a conflict in the event of $firework$ when the light-agent believes it to be a $fire$.

The light-agent, however, does not realize its inferred cause is incorrect as there is no prediction error because the light intensity due to $fire$ and $firework$ are very similar in its generative model (i.e. they share some values of x), as shown in Figure 3.4.

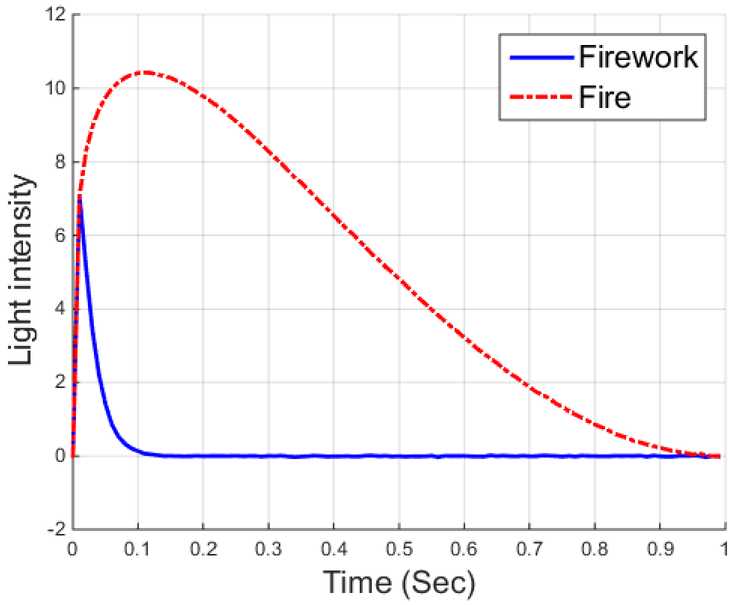


Fig. 3.4: Observations of light-agent for $firework$ (x=0.25) and $fire$ (x=4) are shown. The initial duration of these events generate the same observation and the agent fails to distinguish between them.

Having the model of heat-agent, the light-agent anticipates the sampling frequency of heat-agent to increase to the range that it should be for the case of $fire$ ($x = 4.1$). However, it is surprised as the heat-agent's behavior does not match its expectation. Light-agent initiates communication to minimize its prediction error. The results are shown in Figure 3.5. It can be seen that the light-agent revised its belief for $firework$ since the heat-agent is more confident about its inference (based on the precision, σ_{p_2}). Since communication occurs both ways, the belief of heat-agent is slightly increased. However, it still remains in the range of $firework$. Communication occurs both ways because the conflict is in the minds of both agents (i.e. the heat-agent also did not predict the message from light-agent and is surprised). The agents continue exchanging

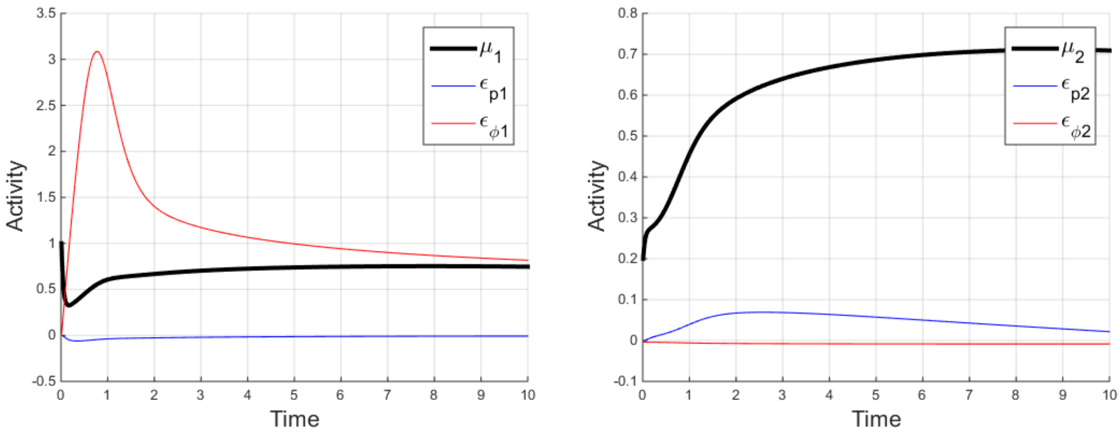


Fig. 3.5: Inference of light-agent (left) and heat-agent (right) regarding $firework$, after communication. The light-agent's inference is improved (it is in the range of $firework$).

messages until the conflict is resolved.

With time, the light intensity due to $fire$ and $firework$ start to differ. Temperature due to heat changes slower than light. Based on these, we construct a scenario of the events $\{noEvent, firework, noEvent, fire\}$, each for 100 seconds duration. Light intensity and temperature observations are shown in Figure 3.6. The final inferences (after settling down), independently and after mutual communication, are shown in Figure 3.7. Before communication, the agents fail in two ways: 1) when $firework$ starts, the light-agent infers the cause of its observation incorrectly as $fire$ ($\mu_1 > 1$), and 2) the heat-agent infers the cause of its observation as $fire$ with a significant delay (at time 340, when the $fire$ had started at 300). Both the issues are resolved after communication and their predictions are in the correct ranges. The incorrect inference by the light-agent is resolved as discussed in the current section just after Figure 3.4. The delay for heat-agent is resolved as follows. The light-agent detects the change earlier and increases its sampling frequency. The heat-agent is surprised by this unexpected change in light-agent's behavior as the former has not detected any significant change in temperature yet. So the heat-agent initiates communication asking the light-agent for the cause of its change in behavior (i.e. the heat-agent samples the light-agent's internal model to

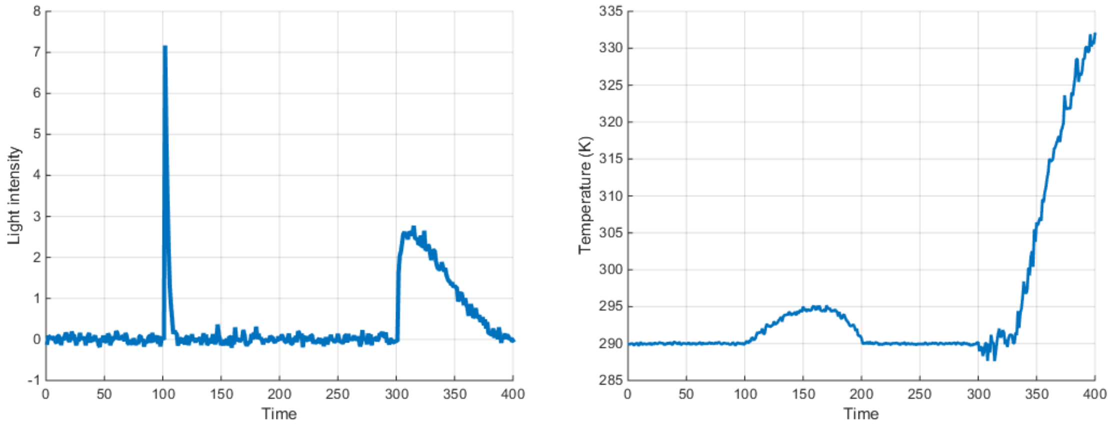


Fig. 3.6: Light intensity (left) and temperature (right) for the simulated scenario of $\{noEvent, firework, noEvent, fire\}$.

minimize its prediction error). The light-agent responds by informing about a significant change in its belief. The conflict is resolved via communication since the heat-agent has learned to trust the light-agent in this situation where the light-agent has high precision (low variance).

3.5 Summary

A novel computational model of distributed decision making is proposed. We show that communication helps a community of predictive coding agents, each limited in its sensorimotor system, to come up with a decision quickly and accurately regarding the state of their shared environment which is not possible for any agent operating independently. The key to this efficiency and accuracy is communication which initiates when a conflict is detected in the mind of an agent due to an error in predicting the other agent's behavior. The proposed model can be scaled to a large number of predictive coding agents operating in a shared environment.

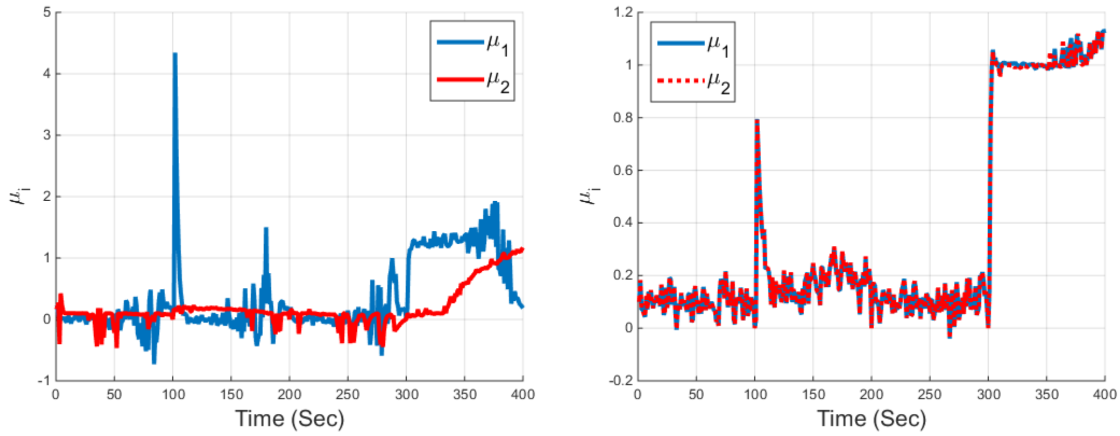


Fig. 3.7: Inference without (left) and with (right) communication. The blue and red lines show the belief of light and heat agents respectively. Conflicts are resolved after communication. [Best viewed in color.]

CHAPTER 4

When to communicate

4.1 Introduction

Activity recognition from sensor data is a core component in many applications. Unfortunately, no single sensor modality can cope with the myriad of real-world situations. While multimodal data is more informative, it has its own challenges [EAG17]. Existing models for multimodal action recognition largely belong to one of two categories: early (data or feature level) or late (decision level) sensor fusion [CJK17]. Limitations of the former method include curse of dimensionality [KMGS14], requirement for time synchronization and same representation for all modalities [PCBH17]. The latter method fuses unimodal decision values from multiple classifiers, ignoring low-level interactions between modalities. It allows different models for different modalities which leads to flexibility, especially when data from any of the modalities is missing [BAM18]. The computational bottleneck of this method is the slowest classifier which delays the final outcome for each data point.

We claim that these limitations can be overcome by replacing the late fusion mechanism by opportunistic communication by each sensor with the other sensors. Consequently, each sensor will estimate the state of the environment generating its observations. In this chapter, we consider the application problem of human activity recognition as a testbed to showcase the effectiveness of our proposal. We model each sensor as an agent in the *predictive coding* framework. Each agent independently estimates the state of its environment and communicates with the other agents only when it is not confident about its estimation. Each agent adapts to its environment over time by unsupervised learning.

Using controlled experiments, we show that limitations of each sensor (agent), such as inference inaccuracy and delay, can be overcome through communication with the other sensors (agents) in our model. Our experiments with the model applied to UTD-MHAD [CJK15b] and MHEALTH [BGHT+14] action recognition dataset show that opportunistic communication and unsupervised adaptation produce near state-of-the-art accuracy. When applied to freezing of gait (FoG) recognition from wearable acceleration sensors in Parkinson's disease (PD) patients [BPR+09], the model produces fast and accurate results. We also show that communication can improve the estimate by each agent; however, full communication does not necessarily lead to better performance, consistent with findings in the literature. The model is also tested for tolerance to sensor failures and mean inference time variation as the number of sensors grows. To the best of our knowledge, this is the first work taking advantage of opportunistic communication by a predictive coding agent for estimating the state of its environment. This has two advantages: efficiency, which is crucial for real-time monitoring; and unsupervised adaptation, which is useful for individualized monitoring and monitoring progressive disorders.

4.2 Related work

Action recognition from video [VLS18, YQL16, LGG+18], inertial sensors and depth cameras [CJK17] has been investigated in the past decade. Motivations of using multiple modalities are studied in [GAGF17]. In multimodal action recognition, information from different sources are fused. Existing techniques can be divided into data-level, feature-level and decision-level fusion [CJK17]. Data-level fusion [LCJK14] combines raw data from different sensors. In these techniques signals should be in the same resolution and format otherwise preprocessing is needed for time synchronization [PCBH17]. Feature-level fusion [KTKL14] merges the features extracted from all modalities to create a high-dimensional feature vector as input to a classifier. Simply concatenating the features is not efficient while finding the most significant feature subset is a challenging task and requires large training sets [GAGF17].

In decision-level fusion, the decisions made by individual classifiers are fused. Commonly used decision-level fusion techniques for action recognition are summation, majority voting, meta-classifiers, etc. [GAGF17, DK18]. Ravi et al. [RDML05] used a plurality voting technique to select the class that has been predicted by a majority of the base classifiers. Dempster-Shafer theory [CJK15a], Highest rank, Borda count, and logistic regression [WLTS06] are also used to weigh the decisions made by the different classifiers. Even though these approaches use a centralized fusion [GAGF17], decision-level fusion allows decentralized processing [BMTH11] which provides a more practical solution to near real-time systems [GAGF17].

The other problem with aforementioned works is that the models are trained offline. It means they cannot adapt for the dynamic changes in the environment. There are online models [NS12, KRS14] for action recognition using single modality. However, these techniques require labels for each data point for adaptation process. In this work, each sensor is modeled as an independent agent in the predictive coding framework. The model can adapt itself in an unsupervised manner. The agents can decide when and how long to communicate. Communication overhead is minimized using a low rank approximation on previous successful communications. The model can also deal with missing values and sensor failures.

4.3 Definitions

Important variables used in this chapter are listed in Table 5.1.

Table 4.1: Symbols and notations for Chapter 4.

Variable	Description
M	Number of variables in the modality
I	Number of states
$\vec{\varphi} \in \Re^{2M}$	Feature vector
$\vec{\mu} \in \Re^{I}$	Belief (expectation of state)
$\vec{\epsilon}_{\varphi} \in \Re^{2M}$	Sensory prediction error
$\vec{\epsilon}_{p} \in \Re^{I}$	Prior prediction error
$\sigma_{\varphi} \in \Re$ $\mathbf{\Sigma}_{\boldsymbol{\varphi}} \in \Re^{2M \times 2M}$	Variance and covariance of generative density
$\sigma_{p} \in \Re$ $\mathbf{\Sigma}_{\boldsymbol{p}} \in \Re^{I \times I}$	Variance and covariance of prior density
$\vec{v} \in \Re^{I}$	Environmental states
$\vec{v}_{p} \in \Re^{I}$	Mean of prior density
N	Number of data points in the dataset

4.4 Models and methods

A typical multimodal action recognition model has five main components [CJK17]: preprocessing, action segmentation, feature extraction, classification and fusion. The proposed agent model for state estimation in this chapter consists of four components: feature extraction, state estimation, communication, and learning. The key advantage of this model is that it allows an agent to decide when to communicate and how long to communicate, which contribute to efficient operation. Communication overhead is minimized using a low rank approximation. Learning allows for adaptation over time. The model can also deal with sensor failures.

4.4.1 Feature extraction

In our model, features are learned from the data using shift-invariant sparse coding because: (1) unlike hand-crafted features that may entangle and hide different explanatory factors of variation behind the data, learned features can adapt to the dataset [BCV13], and (2) the systematic approach to learning features from time-series data is by using shift-invariant similarity [KB18b]. Furthermore, in action recognition, the time-series have different length while beginning and ending of an action is not specified. Shift-invariant sparse coding is useful in the action recognition task since it does not require any pre-processing and data segmentation. Let $\vec{x} \in \Re^q$ be a data point and $\mathbf{D} = [\vec{d}_1, ..., \vec{d}_c] \in \Re^{r \times c}$ be a dictionary of c features, $q \geq r$. The convolution of two signals, $\vec{x}$ and $\vec{d}_j$, is: $(\vec{x} * \vec{d}_j)^{(\tau)} \equiv \sum_{t=1}^{r} \vec{x}^{(\tau)} \vec{d}_j^{(t-\tau)}$, where $*$ is the convolution operator, τ represents a shift in the positive direction, $\tau = 1, 2, ..., q$, and $\vec{x} * \vec{d}_j \in \Re^{q+r-1}$. The optimal shift for the best match between $\vec{x}$ and $\vec{d}_j$ is computed as:

$$\tau_o = \underset{\tau=1,2,...,q}{\operatorname{argmax}} \, (\vec{x} * \vec{d}_j)^{(\tau)} \tag{4.1}$$

Then, the shift-invariant similarity (or coefficient) between a data point $\vec{x}$ and a feature $\vec{d}_j$ is: $\alpha_j = \sum_{t=1}^{q} x^{(\tau_o)} \vec{d}_j^{(t-\tau_o)}$.

Given $\vec{x}$, its best-matching feature in $\mathbf{D}$ and the corresponding coefficient are:

$$\vec{d}_o = \underset{\vec{d}_j \in \mathbf{D}}{\operatorname{argmax}} \, \alpha_j, \qquad \alpha_o = \sum_{t=1}^{q} x^{(\tau_o)} \vec{d}_o^{(t-\tau_o)} \tag{4.2}$$

For shift-invariance, a data point $\vec{x}^{(t)} \in \Re^q$ is modeled as:

$$\vec{x}^{(t)} = \sum_{i=1}^{c} \vec{d}_{o,i}^{(t-\tau_o)} \alpha_{o,i} \tag{4.3}$$

Shift-invariant sparse coding (SiSC) is used in this work. Given a dataset $\mathbf{X} \in \Re^{q \times N}$, SiSC is formulated as minimizing:

$$\ell_{SiSC}(\mathbf{X}) = \frac{1}{2} \sum_{\vec{x}^{(t)} \in \mathbf{X}} \|\vec{x}^{(t)} - \sum_{i=1}^{c} \vec{d}_{o,i}^{(t-\tau_o)} \alpha_{o,i}\|_2^2 \quad s.t.$$

$$\|\vec{\alpha}_o\|_0 \leq \kappa \ \ \forall \vec{x}^{(t)} \in \mathbf{X}, \ \ \|\vec{d}_j\|_2 = 1 \ \ \forall j \tag{4.4}$$

$\alpha_{o,i}$ is computed using Equation 4.2 and matching pursuit [MZ93] from $\mathbf{D}$ and $\vec{x}^{(t)}$. On applying stochastic gradient descent with learning rate η, the dictionary update rule can be derived as:

$$\vec{d}_{o,i} \leftarrow \vec{d}_{o,i} + \eta \sum_{t=1}^{q} (\vec{x}^{(t)} - \hat{\vec{x}}^{(t)}) \alpha_{o,i} \tag{4.5}$$

Each modality may be multivariate. The sequence of indices of the optimal feature and optimal shift for each variable in the modality constitutes the sensory data vector, $\vec{\varphi}$, which is used for state estimation by an agent.

$$\vec{\varphi} = [o_{\vec{d}}^1, o_{\tau}^1, o_{\vec{d}}^2, o_{\tau}^2, ..., o_{\vec{d}}^M, o_{\tau}^M]^T \tag{4.6}$$

where M is the number of variables in the modality (for example, $M = 3$ for an acceleration sensor with three axes) and $\{o_{\vec{d}}^m, o_{\tau}^m\}$ correspond to the m^{th} variable. $o_{\vec{d}}$ and o_{τ} denote optimal feature and optimal shift, respectively.

4.4.2 State estimation by an agent

Each sensor is modeled as an agent in the predictive coding framework. An agent can independently infer the environmental states by minimizing the free energy F [Fri05].

$$F = \int -q(\vec{v}) \ln p(\vec{v}, \vec{\varphi}) d\vec{v} + \int q(\vec{v}) \ln q(\vec{v}) d\vec{v} \tag{4.7}$$

where the first term is the average energy and the second term is negative of entropy associated with the recognition density. Assuming $q(\vec{v})$ to be a sharply peaked Gaussian density function (i.e. the Gaussian bell shape is squeezed towards a delta function), the most likely value of the environmental state is estimated iteratively using Bayesian approximation as [Fri05]:

$$\frac{\partial F}{\partial \vec{\mu}} = \dot{\vec{\mu}} = -\vec{\epsilon}_p + \frac{\partial g(\vec{\mu})^T}{\partial \vec{\mu}} \vec{\epsilon}_\varphi \tag{4.8}$$

where $\vec{\epsilon}_\varphi$ and $\vec{\epsilon}_p$ are updated as:

$$\dot{\vec{\epsilon}}_p = \vec{\mu} - \vec{v}_p - \boldsymbol{\Sigma_p} \vec{\epsilon}_p \tag{4.9}$$

$$\dot{\vec{\epsilon}}_\varphi = \vec{\varphi} - g(\vec{\mu}) - \boldsymbol{\Sigma_\varphi} \vec{\epsilon}_\varphi \tag{4.10}$$

and the prediction errors are $\vec{\epsilon}_\varphi = \boldsymbol{\Sigma_\varphi^{-1}}(\vec{\varphi} - g(\vec{\mu}))$ and $\vec{\epsilon}_p = \boldsymbol{\Sigma_p^{-1}}(\vec{\mu} - \vec{v}_p)$. $P(\vec{\varphi}, \vec{v})$ is initialized using a limited number of samples (e.g. data of another individual) where $\vec{v}$ is state of the environment. The belief is a vector, $\vec{\mu}$, which is the posterior probability distribution over all the states. This distribution is used as a measure of confidence of the agent's inference.

4.4.3 Communication with other agents

In the proposed model, an agent initiates communication when it is not confident about its belief.

$$\text{Confident}(A_j) = \begin{cases} True, & \text{if } max(\vec{\mu}_j) > d \\ False, & \text{otherwise} \end{cases} \tag{4.11}$$

where d is a threshold and $\vec{\mu}_j$ is the j^{th} agent's belief vector. The confidence level influences an agent's decision to communicate. When threshold $d = 1$, the agent is confident of its inference only if the posterior probability is unity which never happens in reality and hence the agent always communicates (a.k.a. *full communication*). When $d = 0$, the agent never communicates. This threshold can be fixed based on sensitivity of the application or using cross-validation on the training data.

Consider $\mathcal{M} \in \Re^{I\times(J+1)\times K}$ where I is the number of states, J is the number of agents and K is the number of previous communication signals (single agent's beliefs) stored in memory. The second dimension of $\mathcal{M}$ is increased by unity to store the true posterior that needs to be inferred. That is, the environment is also considered as an agent; the other agents communicate to discover its state. A few data points are used for learning the priors. The true posteriors corresponding to these data points are given while those for the rest of the data points are missing. In our experiments, it is assumed that multiple classes cannot co-occur at the same time. So the true posteriors constitute a one-hot belief vector. Many entries in $\mathcal{M}$ will be missing if the agents rarely communicate with each other. Whenever an agent initiates communication, its belief is missing and the other agents send their inferred beliefs. Otherwise, the other agents' beliefs are missed. For instance, for the k^{th} data point, the agent A_j either stores its belief in $\mathcal{M}$ or communicates with $A_{j'}$ as follows:

$$\begin{cases} \mathcal{M}_{:,j,k} = \vec{\mu}_j, \Omega_{:,j,k} = \vec{1}, & \text{if Confident}(A_j) = True \\ \mathcal{M}_{:,j',k} = \vec{\mu}_{j'}, \Omega_{:,j',k} = \vec{1}, & \text{otherwise} \end{cases} \tag{4.12}$$

where $\Omega \in \{0,1\}^{I\times(J+1)\times K}$ includes zeros for the missing entries and ones for the given entries in $\mathcal{M}$. The problem of inferring the missing posteriors from a set of observed (communicated) beliefs can be solved using tensor completion with low rank assumptions. Such methods require less training data and are more generalizable [WST14, YLL17] due to fewer training parameters [DZB+14]. The tensor completion problem with low rank assumption is formulated as [LMWY13]:

$$\min_{\mathcal{X}} \; rank(\mathcal{X}) \;\; s.t. \;\; \mathcal{X}_\Omega = \mathcal{M}_\Omega \tag{4.13}$$

where $\mathcal{X}$ and $\mathcal{M}$ are identical size tensors. $\mathcal{X}$ is the completed tensor. Computing the rank of a tensor is an NP-hard problem [CHL13] which is solved by defining the trace norm of tensor [LMWY13]:

$$\min_{\mathcal{X}} \; ||\mathcal{X}||_* \;\; s.t. \;\; \mathcal{X}_\Omega = \mathcal{M}_\Omega \tag{4.14}$$

where $||.||_*$ denotes the trace norm. A fast low-rank tensor completion (FaLRTC) algorithm [LMWY13] is used for solving the optimization problem in Equation 4.14 to estimate the true posteriors from the communicated beliefs. Despite being faster, FaLRTC has a reasonable accuracy compared to other tensor completion methods [LMWY13]. The convergence rate of this algorithm is guaranteed to be $O(n^{-2})$ where n is the number of iterations.

Upon recovering the missing entries in the tensor, the agent A_j, who initiated the communication, infers the state of the environment for the k^{th} data point as follows:

$$\hat{\vec{\mu}}_j = \begin{cases} \vec{\mu}_j, & \text{if Confident}(A_j) = True \\ \mathcal{X}_{:,J+1,k}, & \text{otherwise} \end{cases} \tag{4.15}$$

A priority list is given to the agents to decide with whom to communicate next. This list can be prepared using some metrics such as efficiency or trustworthiness of agents or can be learned using computational models [PCT+10]. In this section, efficiency is considered as a metric whereby the agent who responds faster has a higher priority. Low rank assumption helps to learn the global structures in the data. In each agent, it helps to learn the model of other agents which compensates for their imperfections in the communicated beliefs.

4.4.4 Updating the agent model

After each communication episode, an agent updates its model in an unsupervised manner by minimizing the variational free energy. Communication occurred because the agent failed to estimate the state confidently. This means, the observation from the environment was surprising to the agent. It will learn from the salient observation using active inference. An agent starts with a simple and imprecise model of the environment and improves it via sampling and communication. The priors and generative model are updated as [Bog17]:

$$\frac{\partial F}{\partial \vec{v}_p} = \dot{\vec{v}}_p = \boldsymbol{\Sigma}_{\boldsymbol{p}}^{\boldsymbol{-1}}(\hat{\vec{\mu}} - \vec{v}_p) = \vec{\epsilon}_p \tag{4.16}$$

$$\frac{\partial F}{\partial \boldsymbol{\Sigma_p}} = \dot{\boldsymbol{\Sigma}}_{\boldsymbol{p}} = \frac{1}{2}(\vec{\epsilon}_p \vec{\epsilon}_p^{\,T} - \boldsymbol{\Sigma_p^{-1}}) \tag{4.17}$$

$$\frac{\partial F}{\partial \boldsymbol{\Theta}} = \dot{\boldsymbol{\Theta}} = \vec{\epsilon}_{\varphi} \hat{\vec{\mu}}^{T} \tag{4.18}$$

The index j is ignored as all agents follow the same update rules. Here $\boldsymbol{\Theta} \in \Re^{2M \times I}$ (parameters of the generative model) contains the mean of all observations for each state.

The proposed model for *when to communicate* summarized in Algorithm 2.

4.5 Experimental results

The proposed model is used in two experiments: (1) human action recognition, and (2) recognition of FoG in PD patients.

4.5.1 Human action recognition

Here the model is evaluated for human action recognition.

Data. UTD-MHAD [CJK15b] is a multimodal action dataset, captured by a Microsoft Kinect camera and a wearable inertial sensor. This dataset contains 27 actions performed by eight subjects. Each subject performed each action four times. The modalities in this dataset are Kinect skeleton, RGB videos, depth videos and inertial signals. After removing three corrupted sequences, the dataset includes 861 sequences.

MHEALTH [BGHT+14] dataset has three inertial measurement units (IMU) placed on 10 subjects' chest, right wrist, and left ankle while they were performing 12 activities.

Algorithm 1 Pseudo code for state estimation by an agent via communication with other agents.

Input Environmental signals $\mathbf{X}$, a priority list P of agents.
Output Belief $\hat{\vec{\mu}}$ over states.
Initialize model parameters $\{\vec{v}_p, \mathbf{\Sigma_p}, \mathbf{\Theta}\}$ using a few environmental signals $\mathbf{X}' \not\subset \mathbf{X}$, $\mathbf{D} \leftarrow SiSC(\mathbf{X}')$ [ref. Sec. 4.4.1, learning dictionary of features].

Agent A_j gets a new signal $\vec{x} \in \mathbf{X}$ from its environment
$\vec{\varphi} \leftarrow [o^1_{\vec{d}}, o^1_{\vec{\tau}}, o^2_{\vec{d}}, o^2_{\vec{\tau}}, ..., o^M_{\vec{d}}, o^M_{\vec{\tau}}]^T$
***** State estimation by the agent A_j *****
$\vec{\mu}_j \leftarrow$ EstimateState($\vec{\varphi}_j$) [ref. Sec. 4.4.2]
if Confident(A_j) **then**
    $\mathcal{M}_{:,j,k} \leftarrow \vec{\mu}_j$
    $\Omega_{:,j,k} \leftarrow \vec{1}$
    $\hat{\vec{\mu}}_j \leftarrow \vec{\mu}_j$
else
    count $\leftarrow 0$
    while Confident(A_j)=$False$ and count$< |P|$ **do**
        count $\leftarrow$ count + 1
        $j' \leftarrow P$[count]
        $\vec{\mu}_{j'} \leftarrow$ EstimateState($\vec{\varphi}_{j'}$) [ref. Sec. 4.4.2]
        $\mathcal{M}_{:,j',k} \leftarrow \vec{\mu}_{j'}$
        $\Omega_{:,j',k} \leftarrow \vec{1}$
        $\mathcal{X} \leftarrow$ FaLRTC($\mathcal{M}, \Omega$) [ref. Sec. 4.4.3]
    end while
    Update model $\{\vec{v}_p, \mathbf{\Sigma_p}, \mathbf{\Theta}\}$ [ref. Sec. 4.4.4]
    $\hat{\vec{\mu}}_j \leftarrow \mathcal{X}_{:,J+1,k}$
end if
Return $\hat{\vec{\mu}}_j$

The modalities are accelerometer, gyroscope, magnetometer and ECG.

Experimental setup. For experiments with UTD-MHAD, depth, skeleton and inertial signals are used in this work. The frame size in depth data is reduced by a factor of 10 to help depth agent's efficiency. The agents' generative models (gaussian density parameters) are learned using data of four subjects (subjects one to four) which are excluded from rest of the experiments. These subjects were considered in [CJK15b] as training set and using them to train the generative models allows appropriate comparison. First, a dictionary of features are leaned on data of these subjects using SiSC algorithm for each modality, separately.

The inference is done for the first agent, independently by considering the index of the best matched feature for each variable in the modality (e.g. inertial sensor has six variables) and its corresponding optimal shift as input, and the posterior probability distribution over all possible states (action categories) as output. Comparing the probability of the most likely state with the desired confidence level (see Table 4.2), the agent decides whether to communicate or not. In the former case, the tensor will be updated by the belief of the next agent in the priority list while the belief of others are missing. This means the second agent also needs to independently estimate the state. Otherwise, the belief of the current agent will be updated in the tensor while others are missing. After tensor completion, the agents decide if further communication is needed, i.e. whether the probability reached the desired level. Further communication means the inference using the third agent also needs to be done and sent to the corresponding entry of the tensor. Whenever there is communication, the internal model of the communicated agents are updated based on the final estimation. The sequence of agents is decided based on their dimensionality from low to high (inertial, skeleton and depth, respectively) to reduce the computational cost. For experiments with MHEALTH, each IMU sensor is considered as an agent. The inference is always initiated with the chest agent since it has the lowest dimensions. Wrist and ankle agents have the same dimensionality so any of them is given precedence over the data. The confidence level is set to $d = 0.9$ based on 10-fold cross validation on data of the four excluded subjects which were used for initializing the generative model.

Performance evaluation. Performance of the model is shown in terms of accuracy. The number of times the agents communicated and the corresponding accuracy for multiple values of the threshold (Table 4.2) shows the benefits of communication. More communication tends to yield higher accuracy but full communication does not necessarily guarantee the highest accuracy, consistent with findings in the literature [WHJ14].

Table 4.2: Accuracy on UTD-MHAD dataset for different thresholds. As threshold increases, communication frequency increases. Comm1: number of times inertial sensor communicated with the skeleton agent. Comm2: number of times the first two agents communicated with the depth agent.

Threshold (d)	Comm1	Comm2	Accuracy %
0	0	0	29.4
0.5	17	1	38.5
0.8	205	91	76.16
0.9	305	133	85.76
0.99	336	203	86.28
1	430	430	84.6

A comparison with existing methods (from Table IV of [HLWL16]) shows that the idea of opportunistic communication in the proposed model yields higher accuracy than most existing methods (Table 5.3). Even though the proposed model has significantly less number of learnable parameters than ConvNets [HLWL16], the latter is only slightly (0.69%) more accurate than the former. The same training set consisting of 431 data points is used for ConvNets and our model. ConvNets has in the order of 60 million parameters. The number of learnable parameters in our model is 22,350 which were trained with five million sampled values from the 431 points (see supplemental material for details). For individual monitoring, the accuracy for each subject is shown in Table 4.4. Table 4.5 compares the accuracy of our model on MHEALTH dataset with recent techniques from Table VIII of [JNSS18] and the best performers in [CTCT18].

4.5.2 Recognition of gait freeze

The model is used for recognizing FoG by continuously monitoring PD patients.

Data. The goal of collecting Dephnet dataset was developing a wearable assistant for PD patients with the FoG. FoG manifests as a sudden and transient inability to move

Table 4.3: Comparison of proposed and existing methods for recognizing 27 actions in the UTD-MHAD dataset.

Method	Accuracy %
ELC-KSVD[ZLZ+14]	76.19
Kinect&Inertial[CJK15b]	79.10
Cov3DJ[HTGES13]	85.58
ConvNets[HLWL16]	86.97
Dawar and Kehtarnavaz [DK18]	86.3
Our model	86.28

Table 4.4: Action recognition accuracy for each subject.

Subjects	sub5	sub6	sub7	sub8
Accuracy %	94.44	82.41	88.89	80.19

involving about 50% of all PD patients [BPR+09]. The dataset contains data of 10 PD patients. Three acceleration sensors (each has three variables) were used at ankle, knee and hip of the patients.

Experimental setup. In this experiment, three agents (ankle, knee and hip agents) are used to recognize whether the state is FoG or not (the action can be stand, walk or turn). The first three patients were used to initialize the generative models and excluded from the experiments. The desired confidence level for initiating communication is set to 0.9 based on 10-fold cross-validation on data of the excluded patients. A moving average filter with window length 10 is used to smooth the estimations and reduce the false alarms due to noises.

Performance evaluation. Performance of the model in terms of standard evaluation metrics, such as mean accuracy, sensitivity, specificity, precision and F-measure, is shown in Table 5.4. Comparing with the user-independent results in [BPR+09] (sensitivity of 73.1 and a specificity of 81.6%) the results are improved. Second row of Table 5.4 shows the results for the case of inference without communication. In this experiment, the three

Table 4.5: Comparison of proposed and existing methods for recognizing 12 actions in the MHEALTH dataset.

Method	Accuracy %
Catal et al.[CTPK15]	94.66
Chen and Xue[CJK15b]	88.67
Jiang and Yin[JY15]	51.46
Ha and Choi [HC16]	84.23
Chowdhury et al.[CTCT18]	91.7
Our model	91.11

sensors are same but have different internal models, and hence different inference accuracy, due to being in different locations of the body. The specificity without communication is low because an agent incorrectly recognized FoG as no event 48% of the times. This issue is significantly resolved with communication; hence collecting beliefs from agents monitoring different locations of the body improves recognition of FoG. Individualized adaption is useful for this application due to two reasons: patients have issues in unique parts of their bodies, and PD is a progressive disease.

The mean inference time in this experiment is 3.76 sec. The model was implemented in Matlab on a computer with 3.8 GHz AMD processor, 32 GB RAM and Windows 10 OS. Training set included 1.8 million sampled values while the number of learnable parameters in the model is 3084 (see supplemental material for details).

Table 4.6: Experimental results on Daphnet freezing of gait dataset, shown with and without communication.

	Accuracy	Sensitivity	Specificity	Precision	F-measure
With	86.39	80.17	96.59	97.47	87.98
Without	65.95	82.89	48.26	62.59	71.73

4.5.3 Skeleton based action recognition

The proposed model excels when the number of sensors is large. Benchmark action recognition datasets rarely exceed a few sensors. To evaluate our model for larger number of sensors/agents, we assume each joint in a Kinect skeleton is an independent agent.

Data. UTD-MHAD [CJK15b] Kinect skeleton dataset (ref. 4.5.1) is used for this experiment.The skeleton has 20 3D joints.

Experimental setup. Each joint in the skeleton is modeled as an independent agent. For example, the head joint is an agent observing only its 3D signals and the communication messages from other joints but does not have access to their observations. It is also unaware of the other agents' internal models. Since the agents are almost equally efficient (3D joints have the same number of variables), the priority list is generated randomly and each experiment is repeated 50 times.

Performance evaluation. Performance of the model in terms of accuracy and communication percentage are shown in Figure 4.1a. Accuracy is highest when the agent communicates around 25.5% of the times, on average with 5.1 other agents. Forcing the agent to communicate with all other agents (setting d=1) does not lead to higher accuracy. The mean inference time with increasing number of agents is tested by setting d = 0.9 and varying the maximum number of agents to communicate with. The time increases fast initially but there is no significant change after 5 as the agent reaches the set confidence level and decides not to communicate further (see Figure 4.1b).

The other advantage of the model is that it can adapt itself over time in an unsupervised manner due to using a predictive coding framework for state estimation. Average number of communications for different trial numbers is shown in Figure 4.1c. In UTD-MHAD, each subject performs each action four times. The number of communications is significantly higher in the first trial than in the fourth. The reason is that the agent learns the observed patterns for each individual and there is no need for further communication in the same situation. The model is also tolerant to sensor failures. In Figure 4.1d, the accuracy and communication percentage is shown versus percentage probability of failure for each sensor. The results show that accuracy is unaffected even if each sensor fails with 0.6 probability. However, the communication percentage (inference time) increases because the agent has to communicate with the next agent in the list if the current agent fails.

Our model has similarities with decision-level fusion methods. Such methods typically fuse the decisions made by *all* models (e.g. classifiers). In the worst case, an agent in our model will communicate with all other agents. Hence, its worst case complexity is the same as the complexity of decision-level fusion. The best case complexity of our model occurs when there is no communication which is significantly more efficient than any fusion method. To analyze the average case complexity, note that communication in our model depends on two factors: the confidence threshold (d) of the agent and the sequence of agents to communicate with. Assuming a fixed d and all agents to be equally likely to provide the needed information, an agent in our model is expected to communicate with half of the other agents.

In our implementation, the agents are ranked in increasing order of their signal dimensionality (each agent observes a multidimensional signal). Thus, the agent with lowest signal dimensionality receives highest preference for communication. This strategy reduces the computational cost as, in order to communicate, each agent has to process its signal whose complexity is a function of the signal dimensionality. In decision-level fusion methods, all models have to process their respective multidimensional signals. So our model is more efficient. Experimentally, we have shown that for skeleton-based action recognition using the UTD-MHAD dataset that the agent reaches its confidence level (d=0.9) after communicating with only 5 out of 19 agents (see Figure 4b).

The confidence threshold (d) is used in the model to manage speed-accuracy trade-off [Wic77]. Without a threshold, the communication loop (Figure 1) would halt when further communication does not improve the confidence level (i.e. the belief or a function of the belief such as entropy is converged). However, one can trade-off accuracy for speed. Table 4.7 shows that a significant amount of time can be saved by compromising less than 1% of accuracy.

An appropriate confidence level is typically 0.9 to 0.95 [ZSB+11]. There are also many techniques in the literature to select the threshold such as extreme value theory [SM12], reinforcement learning [LMHPP13], and cross-validation [Pre98]. The cross-validation is used in this experiment. For each dataset, a small portion of data is excluded from rest of the experiments to be used for initializing the parameters (see experimental setup sections for details). Using cross-validation on the excluded data, the value of d which leads to the best accuracy is fixed. Table 4.7 compares the accuracy and running time with two most common decision-level fusion methods [GAGF17]: majority voting and Naive Bayes. For this comparison, only the communication is replaced with the fusion methods in our model. The results show that communication yields higher accuracy and is faster. The model was implemented in Matlab on a PC with 3.8 GHz AMD processor, 32 GB RAM and Windows 10 OS.

Table 4.7: Comparison of % accuracy (Acc) and running time in seconds (Time) as the communication in our model is replaced with majority voting (MV) and Naive Bayes (NB) decision-level fusion.

Dataset	MV		NB		Ours	
	Acc	Time	Acc	Time	Acc	Time
UTD (d=0.99)	50.4	3.9	70.1	4.4	86.3	2.2
UTD (d=0.9)					85.8	1.3
MHEALTH (d=0.9)	79.1	2.6	85.0	2.9	91.1	2.0
MHEALTH (d=0.85)					90.8	1.4
Daphnet (d=0.9)	73.2	4.2	78.5	4.6	86.4	3.8
Daphnet (d=0.8)					86.0	1.9

4.6 Summary

In this chapter, each sensor is modeled as a predictive coding agent which can sample whenever needed from other agents' internal model via communication and learn its internal model in an unsupervised manner. The model evaluated for action recognition and gait freeze recognition using benchmark datasets. The experimental results are comparable to the state-of-the-art even though the proposed model uses significantly fewer parameters. Our experiments showed that communication can improve the estimation of each agent by overcoming inference inaccuracy and delay. However, full communication does not necessarily lead to the best performance, consistent with findings in the literature.

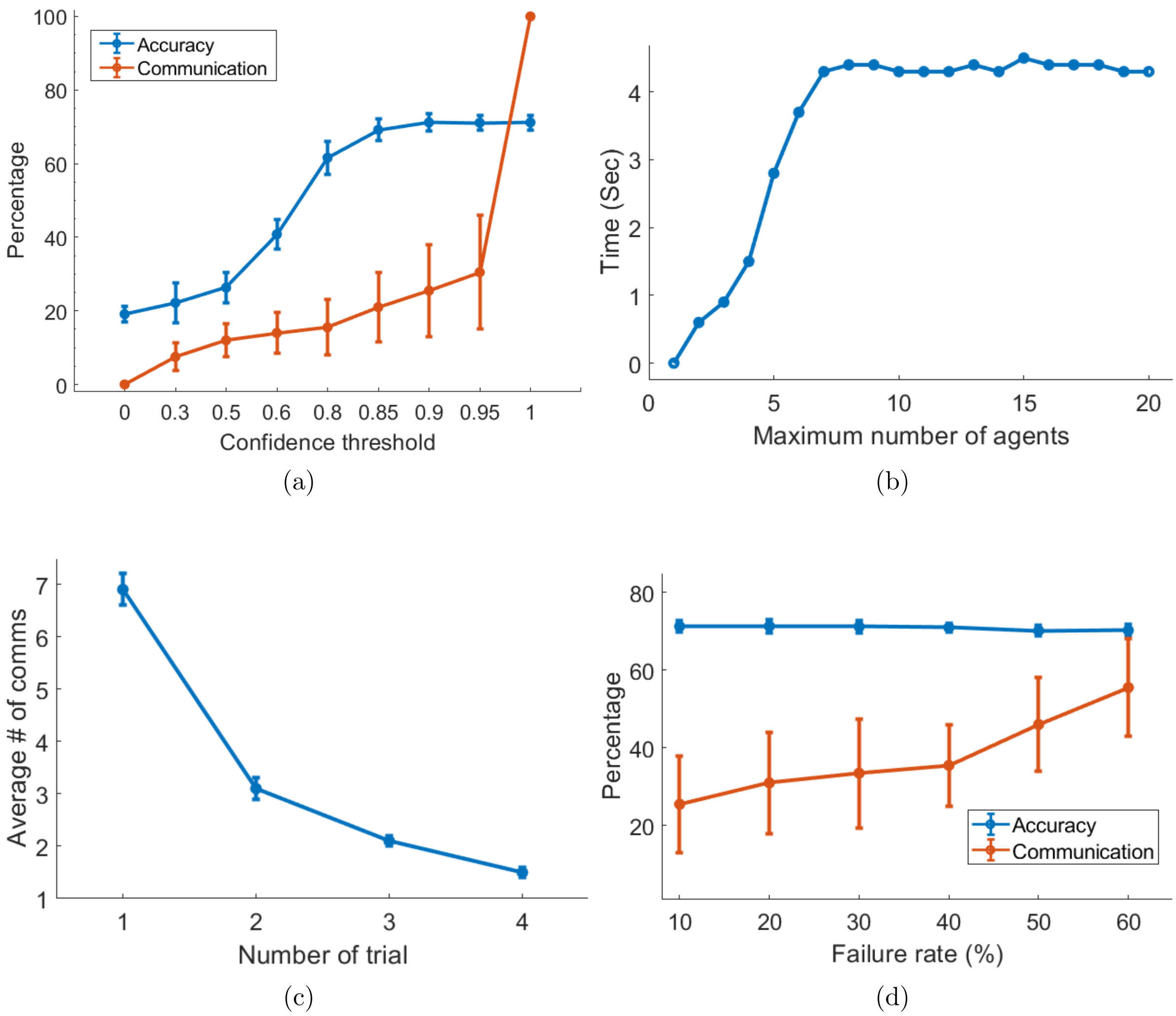


Fig. 4.1: Experimental results on skeleton data: (a) mean and standard deviation of accuracy (blue) and percentage of communication (red) for different levels of confidence, (b) mean inference time when the maximum number of communications is fixed (d=0.9), (c) mean number of communications versus number of trials, (d) mean and standard deviation of accuracy and percentage of communication for different levels of sensor failures (d=0.9).

Chapter 5

With whom to communicate

5.1 Introduction

This chapter investigates how *an* agent can optimally select other agents to communicate with for predicting the state of its environment. We model communication as an action that facilitates active perception [BAT18] whereby an agent actively and selectively samples (or *communicates with*) other agents. Communication makes causal knowledge acquisition efficient by allowing to: (1) share causal knowledge regarding the same event even though the observations are from different sensors in space, time or modality, and (2) acquire high-level causal knowledge directly from another agent instead of from the low-level sensory environment. Hence, communication by an agent is inevitable for predicting its environmental state efficiently.

Learning with whom to communicate is crucial. Full communication does not scale well with the number of agents [Hos17] while predefined protocols cannot adapt to environmental changes or capture dynamic changes in the agents' interactions [HYZW18]. Not all agents are equally informative in a situation. Communication with a less-informative agent increases cost and might reduce the agent's confidence and accuracy.

Partially observable Markov decision processes (POMDPs) have been widely used to learn a state-to-action mapping, referred to as *policy*, which requires a reward function dependent on the agent's goal. Predictive coding [Fri10] is a more general framework for modeling an agent, with no explicit reward function [FDK09]. We propose an agent model in the predictive coding framework with a unified objective (minimization of variational free energy) for inference, learning, and action. Using the same objective, our agent learns a communication policy as a mapping from its belief state to *with whom to communicate.*

5.2 Related work

Learning communication policies allows an agent to autonomically and independently decide whom to communication with. This is particularly beneficial where a neighbor might not be geographically close to the agent, and neighbors change with space, time or the agent's goals. Learning communication protocols with respect to *whom to communicate with* has been limited.

In **distributed AI and multiagent systems**, some recent papers investigated the importance of learning optimal communication targets. Learning an effective communication protocol is a key problem in multiagent systems to solve cooperative tasks in partially-observable environments [DGR+18]. Recent research has highlighted the advantages of targeted interaction over broadcasting messages to all participating agents [DGR+18]. Full communication does not scale well with the number of agents [Hos17]. Predefined protocols cannot adapt to environmental changes and capture dynamic changes in agents' interactions [HYZW18]. Furthermore, not all the agents are equally informative in a situation, so communication with a non-optimal agent might reduce agent's confidence and accuracy [KB19].

Relational reinforcement learning is used in [PCT+10] in which the environment is fully observable. Hoshen [Hos17] proposed a model that uses CommNet [SF+16] with an attention mechanism whereby a central controller decides which agents to communicate with (i.e. communication policies are globally learned). It lacks the ability to handle heterogeneous agent types [PYW+17] and only considers fully supervised one-step prediction tasks. Das et al. [DGR+18] proposed a decentralized POMDP for planning over long time horizons. While this work allows decentralized execution of policies, during training, a centralized approach guides the optimization of individual agent policies. Using POMDPs requires a reward function. When perception is the goal but not subtask, the reward function should be defined to penalize uncertainty in agent's belief. This reward function can violate piecewise-linear and convex property of the value function required by most POMDP planners [SWOS18]. Defining a reward function that penalizes extensive communication when the internal models of other agents are not observable or known a priori, and can change over time, is nontrivial. Also, the computational cost of POMDP planning grows exponentially with number of agents [SWOS18].

5.3 Definitions

This section introduces the relevant terms and concepts. Throughout this chapter, the random variables and their outcomes are presented by uppercase (X) and lowercase (x) letters, respectively. Vector sign ($\vec{x}$) denotes column vectors.

Definition 1. (Agent) An agent is anything that can perceive its environment through sensors and act upon that environment through actuators [RN16]. The agent estimating its environmental state will be referred to as the *primary agent*.

Definition 2. (Markov decision processes) [RN16] Sequential decision problems in uncertain environments, also called Markov decision processes (MDPs) are defined as tuple [RN16]: $\langle \Psi, A, T_a, r_a \rangle$ where Ψ is a finite set of states, A is a finite set of actions. $T_a(\psi'|\psi, a) = P(\{\Psi_{t+1} = \psi'|\Psi_t = \psi, A_t = a\})$ is the transition probability. r_a is the reward received at state ψ'. The goal is to find a policy $\pi : \Psi \rightarrow A$ that maximizes the cumulative rewards. The objective of MDP can be expressed as the Bellman optimality

Table 5.1: Symbols and notations for Chapter 5.

Variable	Description
I	Number of states.
J	Number of agents.
$\vec{\varphi}^{(e)} \in \Re^{M}$	Feature vector.
$\vec{\varphi}^{(msg)} \in \Re^{I}$	Communication message.
$\vec{\mu}^{(v)} \in \Re^{I}$	Belief vector about environmental states.
$\vec{\mu}^{(u)} \in \Re^{J}$	Belief vector about control states.
$\vec{\epsilon}_{\varphi^{(e)}} \in \Re^{M}$	Sensory prediction error.
$\vec{\epsilon}_{\varphi^{(msg)}} \in \Re^{I}$	Communication message prediction error.
$\vec{\epsilon}_{p^{(e)}} \in \Re^{I}$	Prior prediction error.
$\vec{v}_p \in \Re^{I}$	Mean of prior density.
$\mathbf{\Theta}_{g_e} \in \Re^{M\times I}$ $\mathbf{\Theta}_{g_{A_{j'}}} \in \Re^{I\times I}$	Parameters for agent's model of environment and other agent $A_{j'}$ respectively.
$\mathbf{\Theta}_{g_\pi} \in \Re^{J\times I}$	Parameters for encoding optimal policy.
$\mathbf{\Sigma}_{\chi}$	Covariances of random fluctuations where $\chi = \{\vec{\varphi}^{(e)}, \vec{\varphi}^{(msg_{j'})}, \pi, p^{(e)}\}$.

equation [Bel52]: $Value(\psi) = r_a + \max_{a\in A} \sum_{\psi'} T_a(\psi'|\psi, a) Value(\psi')$ where $Value(\psi)$ is the utility or value of state ψ.

Definition 3. (Partially observable MDPs) [RN16] Partially observable MDPs (POMDPs) is an extension of MDP when the states are partially observable. A POMDP can be converted to a MDP using beliefs about the current state. The belief can be recursively computed from the observations and actions using Bayes rule. POMDP based approaches can provide a *closed-loop non-myopic* solution for agents' optimal decision making problem [RN16]. Most of existing POMDP solvers are designed for purposes when reducing uncertainty is a subtask and not a goal. They fail for active perception due to requiring a long time for computing policy or underlying assumptions (e.g. piecewise linearity) that do not hold for a belief based reward function required for active perception [SWOS18].

Definition 4. (Recognition density) [Fri10] Recognition density is a probabilistic representation of environmental states which is encoded by internal states μ. Assuming a Gaussian density allows Laplace approximation: $Q(\psi) = \mathcal{N}(\psi;\mu,\zeta) = \frac{1}{\sqrt{2\pi\zeta}} exp(-(\psi - \mu)^2/2\zeta)$.

Definition 5. (Generative density) [FDK09] Generative density $p(\varphi, \psi)$ is a joint probability density relating environmental states and observations. It includes a sensory mapping $\varphi = g(\tilde{v}, \tilde{u}, \theta_g) + \tilde{\omega}_1$ and equation of motions $\dot{\tilde{v}} = f(\tilde{v}, \tilde{u}, \theta_f) + \tilde{\omega}_2$ [FDK09], where $\tilde{\omega}_i (i = 1, 2)$ are Gaussian noise. The latter contains the policies encoded in the parameters θ_f. It is a joint probability distribution over states, control states and the learned parameters. v and u are environmental hidden states and control states, respectively. $\tilde{X}$ shows the generalized coordinates of the variables. We use second order generalized coordinates consisting of state and change of state.

Definition 6. (Sampling probability) [FSM12] Sampling probability $R(\varphi'|\varphi, a) = p(\{\varphi_{t+1} = \varphi'|\varphi_t = \varphi, a_t = a\})$ is the probability that the observation $\varphi' \in \Phi$ follows action $a \in A$ given φ.

5.4 Models and methods

State estimation can be formulated as Bayesian inference [KR96]: $p(\Psi_t|\Phi_{1:t}) \propto p(\Phi_{1:t}|\Psi_t)p(\Psi_t)$. Active perception is defined as [DB02] $p(\Psi_t|A_{1:t}, \Psi_{1:t})$, in which the *previous actions are causes for the current observation.* Such problems are traditionally solved by POMDPs for closed-loop non-myopic decision-making. We consider other agents as active parts of an agent's environment so that it can change their control states via communication which is an action. The problem is formulated as:

$$p(\Psi_t|A_{1:t}, \Phi_{1:t}) = \frac{p(\Phi_{1:t}|\Psi_t, A_{1:t})p(\Psi_t, A_{1:t})}{p(\Phi_{1:t}, A_{1:t})} \tag{5.1}$$

A number of challenges need to be addressed: (1) The size of action space grows exponentially with the number of agents, rendering standard POMDP solvers infeasible [SWOS18]. In our problem, outcome of action (the received communication message) is not deterministic so the agent should choose the communication targets sequentially and take the new observations into account. For example, the agent should be aware of an agent's failure in providing communication message in order to ask another agent with similar expertise. This makes the size of action space equal to P_k^J where P, J and k denotes permutation, number of agents and number of communications where k is not known a priory (i.e. number of agents to communicate with is not predefined). (2) Since all agents are not equally informative and their internal models are unobservable and time-varying, the problem needs to be solved online, without supervision or reinforcement. (3) An agent has to assign a degree of trust to each message received and update its belief accordingly.

We consider Ψ as a collection of causal environmental states that influences observations. It includes V as the uncontrollable aspects of environment and U which can be controlled by an agent. We model communication as an action using which an agent changes other agents' control states. We distinguish between A and U as an action may fail to control other agents. The action reveals a new observation, communication message $\Phi^{(msg)}$ that depends on U (activated by action) and V. Therefore, the random variable Φ collects two types of observations: $\Phi^{(e)}$ generated by the shared environment and $\Phi^{(msg)}$ generated by other agents as controllable parts of environment. The goal is to infer V at time t, efficiently, by activating the optimal sequences of $U_{1:t}$. Obviously, Φ_t is conditionally independent of action A, given Ψ which consists both U and V. Accordingly, the problem of *with whom to communicate* is converted to inferring the optimal sequence of control states $U_{1:t}$. Rewriting the above discussion as $p(\Psi_{1:t}|\Phi_{1:t})$, the problem is a Bayesian inference where exact computation is intractable for large distributions.

We approximate the posterior belief using variational inference [FR12], by minimizing divergence between a recognition density and the posterior density to reach $D_{KL}(Q(\Psi_{1:t})||p(\Psi_{1:t}|\Phi_{1:t})) = F + \ln p(\Phi_{1:t})$ where F is the VFE in Def. 4. Hence we can formulate our agent's model in the PC framework (Def. 4). We then provide an algorithm for sequentially optimizing perception and action, and updating agents' model as well as optimal policy.

In order to mathematically define optimal information gathering through communication, we formulate it as an active inference framework. The active inference tuple in our model is defined as follows:

- Ψ_t is a random variable that represents the state of the environment at time t. It is divided into hidden environmental states V_t and hidden control states U_t. V is the true state of the shared environment which produces sensory signals. U represents the aspects of the environment that can be controlled. In this work, other agents are considered as part of the environment. Their control states are defined as sending a communication message or not.

- A_t is a random variable which represents the agent's action at time t. Actions can generate new observations by changing control states of the environment. Communication is considered as an action in our model. It can change the controllable states of the other agents U, by asking them to communicate.

- Φ_t is a random variable representing the sensor input (observations) of the agent at time t. Observations are a function of hidden environmental states. In our model, Φ includes both sensory signals from shared environment and communication message from other agents. In fact, communication messages are extra-sensory samples provided to an agent upon request.

- ϑ represents real valued internal states of the agent which parameterize a conditional density (e.g. Gaussian density is parameterized by its mean and standard deviation).

- Generative density $G = p(\Phi_{1:t}, \Psi_{1:t})$ is a joint probability density relating environmental states and sensory data. It can be specified in the form of a likelihood and a prior. In our model, it is defined as:

$$p(\Phi_{1:t}, \Psi_{1:t}) = p(\Phi_{1:t}|\Psi_{1:t})p(\Psi_{1:t}) \tag{5.2}$$

  As in POMDPs, the Markovian observation model implies that the observation at time t depends only on the current environmental state, so the likelihood term is written as:

$$p(\Phi_{1:t}|\Psi_{1:t}) = p(\Phi_1|\Psi_1)...p(\Phi_t|\Psi_t) = \prod_t p(\Phi_t|\Psi_t) \tag{5.3}$$

  The transition probabilities have some differences with POMDPs in the sense that they depend on the parameters ϑ but in general, they have the following form [FSM12]: $p(\Psi_{1:t}) = p(\Psi_0)\prod_t p(\Psi_t|\Psi_{t-1})$. In fact, the prior expectations over trajectory of hidden controlled states are where the optimal policy is incorporated [FDK09]. In other words, the parameters are optimized to represent optimal policy for communication.

- Sampling probability $R = p(\Phi_{t+1}|\Phi_t, a_t)$ is agent's prediction of its action's consequences. That is, the agent needs to learn an internal model of other agents over time so that given its current sensory input, it can predict others' responses to communication. It is worth noting that the real communication message can be different from agent's prediction and the model is updated using prediction error.

- Recognition density $Q(\Psi_{1:t}, \vartheta|\mu_{1:t})$, is an approximate posterior over states and parameters which is encoded with its sufficient statistic $\mu_{1:t}$, in the agent's internal model. The density is assumed to be Gaussian which is a common choice in the literature [Fri10].

The unified objective of each agent for inference (perception), learning and communication (action selection in general) is to minimize the VFE (Def. 5):

$$F = -\int Q(\Psi_{1:t}) \ln p(\Psi_{1:t}, \Phi_{1:t}) d\Psi + \int Q(\Psi_{1:t}) \ln Q(\Psi_{1:t}) d\Psi + C \tag{5.4}$$

For perception and action, the agent solves the following dual optimization, sequentially [FSM12]:

$$\mu_t = \underset{\mu}{\operatorname{argmin}} F(\{\Phi_0, ..., \Phi_t\}|\mu) \tag{5.5}$$

$$a_t = \underset{a}{\operatorname{argmin}} \sum_{\Phi} R(\Phi_{t+1}|\Phi_t, a) F(\{\Phi_0, ..., \Phi_{t+1}\}|\mu_t) \tag{5.6}$$

Finally, when state estimation for each data sample is converged, all the parameters and hyperparameters of model including priors and policy, as well as internal model of environment and other agents are optimized in an online manner for minimizing VFE.

Since $Q(\Psi_{1:t}, \vartheta)$ is a Gaussian, with Laplace approximation, Equation 5.4 converts to [BKMS17]:

$$F = -\ln p(\mu_{1:t}, \Phi_{1:t}) + C \tag{5.7}$$

The first term after equality is the generative density in which the environmental states are approximated by sufficient statistics of recognition density (agent's belief) and C is a constant. An intuitive interpretation of the above equation is that the agent interprets the external states of the environment (including both sensory states and hidden environmental states), in terms of its hidden internal states $\mu_{1:t}$ (see [BKMS17] for a mathematical proof). The agent's ultimate goal is to estimate the first order coordinate of states which is the aspect of the environment intended to be estimated. Inferring changes in the states helps for a more accurate inference but is not the primary goal.

5.4.1 Independent inference by an agent.

In our model, an agent starts with an independent estimation based on its private sensory signals. Real world sensors generate high dimensional noisy time-series. Typically, extracted features from raw data are used in machine learning to achieve higher accuracy in estimating state of the environment [CJK15b]. Hand-crafted features may hide different explanatory factors of variation in the data so unsupervised feature learning is recommended [BCV13]. The systematic approach to learning features from time-series data is by using shift-invariant similarity measure [KB18b]. In this work, features are learned from the data using shift-invariant sparse coding (SiSC). The sparse codes are considered as agents' observations from sensory signal. Given a dataset $\mathbf{X} \in \Re^{q \times N}$, SiSC is formulated as minimizing:

$$\ell_{SiSC}(\mathbf{X}) = \frac{1}{2} \sum_{\vec{x}^{(t)} \in \mathbf{X}} \|\vec{x}^{(t)} - \sum_{i=1}^{c} \vec{d}_{o,i}^{(t-\tau_o)} \alpha_{o,i}\|_2^2 \quad s.t.$$
$$\|\vec{\alpha}_o\|_0 \leq \kappa \;\; \forall \vec{x}^{(t)} \in \mathbf{X}, \;\; \|\vec{d}_j\|_2 = 1 \;\; \forall j \tag{5.8}$$

where $\vec{x}^{(t)} \in \Re^q$ is the sensory signal generated at time t and c is the number of features $\vec{d} \in \Re^r$ in a dictionary of features $\mathbf{D} = [\vec{d}_1, ..., \vec{d}_c] \in \Re^{r \times c}$, $(r \leq q)$. $\vec{d}_o$ represents the best-matching (optimal) feature for signal $\vec{x}$ while τ_o shows optimal shift in time for $\vec{d}_o$. Activation strength of d_o is represented by a real value α_o which is called sparse code or coefficient. $||.||_F$ denotes the matrix Frobenius norm. $\alpha_{o,i}$ is computed using convolutional version of matching pursuit [MZ93] from $\mathbf{D}$ and $\vec{x}^{(t)}$. On applying stochastic gradient descent with learning rate η, the dictionary update rule can be derived as: $\vec{d}_{o,i} \leftarrow \vec{d}_{o,i} + \eta \sum_{t=1}^{q} (\vec{x}^{(t)} - \hat{\vec{x}}^{(t)})\alpha_{o,i}$. For a detailed description of SiSC, refer to[KB18b].

Each sensor may generate multivariate time-series. The sequence of indices of the optimal feature and optimal shift for each variable constitutes the sensory feature vector $\vec{\varphi}^{(e)}$, which is considered as an agent's observation:

$$\vec{\varphi}^{(e)} = [o^1_{\vec{d}}, o^1_{\tau}, o^2_{\vec{d}}, o^2_{\tau}, ..., o^M_{\vec{d}}, o^M_{\tau}]^T \tag{5.9}$$

where M is the number of variables. The superscript e is used to distinguish each agent's private observation of the shared environment from the communication messages $\vec{\varphi}^{(msg)}$. At this time, agent's observations are limited to $\vec{\varphi}^{(e)}$ so the objective function is simplified to:

$$F^{(e)} = -\ln[p(\vec{\varphi}^{(e)}|\vec{\mu}^{(v)})p(\vec{\mu}^{(v)})] + C \tag{5.10}$$

where $p(\vec{\varphi}^{(e)}|\vec{\mu}^{(v)}) = p(\vec{\varphi}^{(e)}|\vec{v}) + \omega_1$ and $p(\vec{\mu}^{(v)}) = \vec{v}_p + \omega_2$. $\vec{\mu}^{(v)}$ denotes the belief vector regarding the aspect of environmental states $\vec{v}$, which should be estimated. Gaussian assumptions about error terms $w_i(i = 1, 2)$, specify likelihood and priors as $\mathcal{N}(\vec{\varphi}^{(e)}; g_e(\vec{\mu}^{(v)}, \mathbf{\Theta}_{g_e}), \mathbf{\Sigma}_{\varphi^{(e)}})$ and $\mathcal{N}(\vec{\mu}^{(v)}; \vec{v}_p, \mathbf{\Sigma}_{p^{(e)}})$, respectively. Mean of likelihood density, $g_e(\vec{\mu}^{(v)}, \mathbf{\Theta}_{g_e}) = \mathbf{\Theta}_{g_e}\vec{\mu}^{(v)}$, is the generative function which maps agent's belief to the environmental observations $\vec{\varphi}^{(e)}$. In this chapter, it is assumed to be a linear function, however, there is no limitation for using non-linear functions [Fri05], as long as they are differentiable. In our model, g_e is initialized using a limited number of samples and updated by observing each new sample in an online manner (details in Sec. 5.4.4). Plugging the Gaussians in Equation 5.10, $F^{(e)}$ is computed as:

$$
\begin{aligned}
F^{(e)} =& -\frac{1}{2}(-\ln|\mathbf{\Sigma}_{p^{(e)}}|- \\
&(\vec{\mu}^{(v)}-\vec{v}_p)^T\mathbf{\Sigma}_{p^{(e)}}^{-1}(\vec{\mu}^{(v)}-\vec{v}_p)-\ln|\mathbf{\Sigma}_{\varphi^{(e)}}|- \\
&(\vec{\varphi}^{(e)}-g_e(\vec{\mu}^{(v)},\mathbf{\Theta}_{g_e}))^T\mathbf{\Sigma}_{\varphi^{(e)}}^{-1}(\vec{\varphi}^{(e)}-g_e(\vec{\mu}^{(v)},\mathbf{\Theta}_{g_e})))+C
\end{aligned}
\tag{5.11}
$$

where $|.|$ denotes determinant of matrix. For a derivation of Equation 5.11 from Equation 5.10, refer to [Bog17].

The best guess can be found by stochastic gradient descent:

$$
\dot{\vec{\mu}}^{(v)} = \frac{\partial F^{(e)}}{\partial \vec{\mu}^{(v)}} = -\vec{\epsilon}_{p^{(v)}} + \frac{\partial g_e(\vec{\mu}^{(v)},\mathbf{\Theta}_{g_e})^T}{\partial \vec{\mu}^{(v)}}\vec{\epsilon}_{\varphi^{(e)}} \tag{5.12}
$$

where $\vec{\epsilon}_{\varphi^{(e)}}$ and $\vec{\epsilon}_{p^{(v)}}$ are auxiliary variables representing $\mathbf{\Sigma}_{\varphi^{(e)}}^{-1}(\vec{\varphi}^{(e)}-g_e(\vec{\mu}^{(v)},\mathbf{\Theta}_{g_e})))$ and $\mathbf{\Sigma}_{p^{(e)}}^{-1}(\vec{\mu}^{(v)}-\vec{v}_p)$, respectively. These terms describe prediction errors weighted by precision (inverse of variance). The former expresses deviation between agent's prediction $g_e(\vec{\mu}^{(v)},\mathbf{\Theta}_{g_e}))$ and actual observation $\vec{\varphi}^{(e)}$, while the latter denotes deviation of estimation $\vec{\mu}^{(v)}$ from prior expectation $\vec{v}_p$. Multiplying with precision terms weigh the influence of each error term in the inference. In other words, these weights define the relative degree of agent's attention to its prior knowledge and current sensory input.

5.4.2 Selecting whom to communicate with.

For each data sample, the agent ought to refine its initial and probably imprecise guess $\vec{\mu}^{(v)}$ through actions. Agents' actions change the control states of the environment, and hence the observations. Since communication is an action, the other agent's message will be an additional observation given that its control state is activated by the primary agent's action (request for communication). In this chapter, we assume that the other agent sends its belief vector as the message. Taking into account the conditional independencies in our model, optimal action is selected as:

$$\begin{aligned} a_t = \underset{a}{\operatorname{argmin}} \sum_{\Phi} \underbrace{p(\vec{\varphi}^{(msg)}_{t+1}|\vec{\varphi}_t, a)}_{1} \Bigg[\ln \underbrace{p(\vec{\varphi}^{(e)}|\vec{\mu}^{(v)}_t)}_{2} + \\ \sum_{\tau=1}^{t} \ln \underbrace{(\vec{\varphi}^{(msg)}_{\tau}|\vec{\mu}^{(u)}_{\tau}, \vec{\mu}^{(v)}_{\tau})}_{3} + \ln \underbrace{p(\vec{\mu}^{(v)}_t)}_{4} + \\ \sum_{\tau=1}^{t} \ln \underbrace{p(\vec{\mu}^{(u)}_{\tau+1}|\vec{\mu}^{(u)}_{\tau}, \vec{\mu}^{(v)}_{\tau})}_{5} \Bigg] \end{aligned} \tag{5.13}$$

where $\vec{\mu}^{(v)}_{t=1}$ is the agent's best guess calculated from Equation 5.12. Equation 5.13 implies agent A_j chooses to communicate with agent $A_{j'}$ $(a = j')$ whom A_j believes would maximally decrease the VFE. The second and fourth terms are defined in the last section, following Equation 5.10. The third term contains model of another agent. An agent needs to learn a model of other agents from their messages, in order to interpret the observations generated by them. This model has the same form as the generative function of environment g_e but with different parameters: $\mathcal{N}(\vec{\varphi}^{(msg_{j'})}; g_{A_{j'}}(\vec{\mu}^{(v)}, \vec{\mu}^{(u)}, \mathbf{\Theta}_{g_{A_{j'}}}), \mathbf{\Sigma}_{\varphi^{(msg_{j'})}})$ where $g_{A_{j'}}(\vec{\mu}^{(v)}, \vec{\mu}^{(u)}, \mathbf{\Theta}_{g_{A_{j'}}}) = \mu^{(u_{j'})}\mathbf{\Theta}_{g_{A_{j'}}}\vec{\mu}^{(v)}$ where $\mu^{(u_{j'})} = 1$ means that control state of $A_{j'}$ is activated by action. The parameters $\mathbf{\Theta}_{g_{A_{j'}}}$, are learned over time by the samples of communication provided by $A_{j'}$ to A_j and are unique for each agent in the environment (details in Sec. 5.4.4).

The fifth term represents agent's prior beliefs about transition among states (equations of motion in Def. 6). It is different from transition function in POMDPs in the sense that it depends on the parameters ϑ. Optimal priors over these parameters make this term equivalent to optimal policy [FSM12]. In other words, $p(\vec{\mu}^{(u)}_{\tau+1}|\vec{\mu}^{(u)}_{\tau}, \vec{\mu}^{(v)}_{\tau}) = T(\Psi_{\tau+1}|\Psi_{\tau}, \pi(\Psi_{\tau}))+ \omega_3 = T(U_{\tau+1}|U_{\tau}, V_{\tau}, \pi(\Psi_{\tau})) + \omega_3$, where V_τ does not change over $\Delta\tau \to 0$ so $V_{\tau+\Delta\tau} \approx V_{\tau}$. Therefore, the fifth term is a Gaussian $\mathcal{N}(\vec{\mu}^{(u)}_{\tau+1}; g_\pi(\vec{\mu}^{(u)}_{\tau}, \vec{\mu}^{(v)}_{\tau}, \mathbf{\Theta}_\pi), \mathbf{\Sigma}_\pi)$. In this model, the next control state $\vec{\mu}^{(u)}_{\tau+1}$ needs to be inferred since the agent should choose the communication target. The agent knows with whom it has already communicated so $\vec{\mu}^{(u)}_{\tau} = \vec{u}_\tau$. Thus it will communicate with $A_{j'}$ only if it has not communicated with it, i.e. $\mu^{(u_{j'})}_{\tau} = u_{j'} = 0$. Therefore, the generative function for trajectory of control states (priors on the dynamics) is defined as:

$$g_\pi(\vec{\mu}^{(u)}, \vec{\mu}^{(v)}, \mathbf{\Theta}_\pi) = (\vec{\mathbf{1}} - \vec{\mu}^{(u)}) \odot (\mathbf{\Theta}_\pi \vec{\mu}^{(v)})$$

where $\vec{\mathbf{1}} \in \Re^J$ and $\odot$ is element-wise product. Finally, the first term in Equation 5.13 is the sampling probability. It allows the agent to predict other agents' behaviors given the current evidences. $\vec{\varphi}^{(msg)}_{t+1}$ is A_j's prediction about the next observation. Therefore, it is not necessarily accurate and may cause prediction errors. This means the agent cannot go inside other agents' mind and verify its prediction [FF15] but it can only predict their response to the best of its ability and after receiving communication message, update its model of that agent using prediction errors (details in Sec. 5.4.4).

5.4.3 Updating belief using communication message.

Section 5.4.2 discussed how A_j selects an agent $A_{j'}$ whom it believes is more knowledgable given the evidences and requests a communication message. It is assumed that receiving communication message is guaranteed upon request (i.e. if $a = j'$, then $u_{j'} = 1$). However, distinguishing action a from control state u and u from its representation in agent's model $\vec{\mu}_{(u)}$ gives the model the flexibility to be used when the current agent does not receive a response.

The new sample $\vec{\varphi}_{t+1}^{(msg)}$ is interpreted through agent's internal model in the same way $\vec{\varphi}^{(e)}$ is processed. This helps the agent to *reason whether it wants to update its belief or not* based on the reliability of the sender. Reliability of $A_{j'}$'s messages are measured by the precision term, $\mathbf{\Sigma}^{-1}_{\varphi^{(msg_{j'})}}$. It also helps in modeling heterogenous agents since they can maintain separate state estimations and see the other agents as an *additional source of information*. The agent's belief is updated by minimizing $F(\{\vec{\varphi}^{(e)}, \vec{\varphi}_1^{(msg)} ..., \vec{\varphi}_{t+1}^{(msg)}\})$:

$$
\begin{aligned}
\dot{\vec{\mu}}_{t+1}^{(v)} = \frac{\partial F}{\partial \vec{\mu}_{t+1}^{(v)}} = -\vec{\epsilon}_{p^{(v)}} + \frac{\partial g_e(\vec{\mu}_{t+1}^{(v)}, \mathbf{\Theta}_{g_e})^T}{\partial \vec{\mu}_{t+1}^{(v)}} \vec{\epsilon}_{\varphi^{(e)}} + \\
\sum_{\tau=1}^{t+1} \frac{\partial g_{A_{j'}}(\vec{\mu}_\tau^{(v)}, \vec{\mu}_\tau^{(u)}, \mathbf{\Theta}_{g_{A_{j'}}})^T}{\partial \vec{\mu}_\tau^{(v)}} \vec{\epsilon}_{\varphi_\tau^{(msg)}} + \\
\sum_{\tau=1}^{t+1} \frac{\partial g_\pi(\vec{\mu}_\tau^{(u)}, \vec{\mu}_\tau^{(v)}, \mathbf{\Theta}_\pi)^T}{\partial \vec{\mu}_\tau^{(v)}} \vec{\epsilon}_\pi
\end{aligned}
\tag{5.14}
$$

where $\vec{\epsilon}_\pi = \mathbf{\Sigma}_\pi^{-1}(\vec{\mu}^{(u)} - g_\pi(\vec{\mu}^{(u)}, \vec{\mu}^{(v)}, \mathbf{\Theta}_\pi)$ and $\vec{\epsilon}_{\varphi^{(msg)}} = \mathbf{\Sigma}^{-1}_{\varphi^{(msg_{j'})}}(\vec{\varphi}^{(msg_{j'})} - g_{A_{j'}}(\vec{\mu}^{(v)}, \vec{\mu}^{(u)}, \mathbf{\Theta}_{g_{A_{j'}}}))$. Since now $t+1$ is the current time thus $\varphi_{t+1}^{(msg)}$ is the current observation and not a prediction. Derivation steps of Equation 5.14 from Equation 5.7 are provided in Appendix A.

5.4.4 Updating the agent's internal model.

Thus far, we have discussed how the agent perceives from and acts on the environment given a generative density. This section explains how the ***generative density itself is learned*** using the same objective function, VFE [Fri10]. Updating parameters of generative density, in an online and unsupervised manner, helps the agent to progressively adapt itself to minimize free-energy on successive exposure to the same stimulus [Fri05]. In our model, after each communication sequence, if the free energy has converged, the agent updates its model. Communication occurs when the agent fails to estimate the state that adequately explains its observation. It learns from the salient observation and use this knowledge in future. An agent starts with a simple and imprecise model of the environment and improves it via sampling and communication. Here we provide the update rules for all parameters and hyperparameters of the model.

Parameters of environment's generative function are updated as [Bog17]:

$$\frac{\partial F}{\partial \mathbf{\Theta}_{g_e}} = \mathbf{\Sigma}^{-1}_{\varphi^{(e)}} (\vec{\varphi}^{(e)} - g_e(\vec{\mu}^{(v)}, \mathbf{\Theta}_{g_e})) \vec{\mu}_T^{(v)^T} = \vec{\epsilon}_{\varphi^{(e)}} \vec{\mu}_T^{(v)^T} \tag{5.15}$$

where superscript T refers to the matrix transpose operation while subscript T stands for the total communication time (i.e. $T = J$ or total number of agents communicated with when $\Delta F < \epsilon$). Model of agent A_j from each agent $A_{j'}$ where $j' \in \{1, ...J\}$ and $j' \neq j$ is updated as:

$$\frac{\partial F}{\partial \mathbf{\Theta}_{g_{A_{j'}}}} = \vec{\epsilon}_{\varphi^{(msg_{j'})}} \vec{\mu}_T^{(v)^T} \tag{5.16}$$

Update rule for priors is reduced to prior errors [Bog17]:

$$\frac{\partial F}{\partial \vec{v}_p} = \mathbf{\Sigma}^{-1}_{p^{(e)}} (\vec{\mu}^{(v)} - \vec{v}_p) = \vec{\epsilon}_{p^{(v)}} \tag{5.17}$$

Parameters of optimal policy after taking each action at time t, where $\mu_t^{(u_{a_{t-1}})} = 1$, is updated as:

$$\frac{\partial F}{\partial \mathbf{\Theta}_\pi} = (1 - \vec{\mu}_{t-1}^{(u)}) \odot \vec{\epsilon}_\pi \vec{\mu}_{t-1}^{(v)^T} \tag{5.18}$$

The update rules for covariance matrices are:

$$\frac{\partial F}{\partial \mathbf{\Sigma}_\chi} = \frac{1}{2}(\vec{\epsilon}_\chi \vec{\epsilon}_\chi^T - \mathbf{\Sigma}_\chi^{(-1)}) \tag{5.19}$$

where χ should be replaced with the $\vec{\varphi}^{(e)}$, $\vec{\varphi}^{(msg_{j'})}$, π and $p^{(e)}$.

Algorithm 2 summarizes the process of Sec. 5.4. The parameters of the model are updated using stochastic gradient descent (SGD) in line 11 of Algorithm 2. Convergence of SGD to a local minimum is guaranteed when the learning rates are inversely proportional to time.

Algorithm 2 Pseudocode for the proposed algorithm.

Initialize g_e and $U = \vec{\mathbf{0}}$
1: **for** Each environmental observation $\vec{\varphi}^{(e)}$ **do**
2: $\quad \vec{\mu}_{t=1}^{(v)} \leftarrow$ EstimateState($\vec{\varphi}^{(e)}$) [ref. Sec. 5.4.1]
3: $\quad$ **while** $\Delta F > \epsilon$ **do**
4: $\qquad t \leftarrow t + 1$
5: $\qquad a_t \leftarrow$ WhomToCommunicate($\vec{\varphi}^{(e)}, \vec{\mu}_t^{(v)}, \vec{\mu}_t^{(u)}$) [ref. Sec. 5.4.2]
6: $\qquad u_j(a_t) \leftarrow 1$
7: $\qquad \vec{\mu}_t^{(v)} \leftarrow$ EstimateState($\vec{\varphi}^{(e)}, \vec{\varphi}_j^{(msg)}, \vec{\mu}_t^{(v)}, \vec{\mu}_t^{(u)}$) [ref. Sec. 5.4.3]
8: $\qquad$ Calculate $\vec{\epsilon}_{\varphi_j^{(msg)}}, \vec{\epsilon}_\pi$ [ref. Sec. 5.4.3]
9: $\quad$ **end while**
10: $\quad$ Calculate $\vec{\epsilon}_{\varphi^{(e)}}, \vec{\epsilon}_p$ [ref. Sec. 5.4.1]
11: $\quad$ UpdateModel($\vec{\epsilon}_{\varphi^{(msg)}}, \vec{\epsilon}_{\varphi^{(e)}}, \vec{\epsilon}_\pi, \vec{\epsilon}_p$) [ref. Sec. 5.4.4]
12: **end for**

Theorem 1 Full communication does not necessarily guarantee highest state estimation accuracy.

Proof 1 *We show that communicating with one more agent might decrease the state es-*

timation accuracy. The goal is to estimate the environmental state from a collection of observations $p(V|\Phi)$. Φ contains environmental observation and communications messages. We approximate the posterior (belief) with Gaussian densities. The goal is to minimize the distance between the true Gaussian $\mathcal{N}_{true}$ and the agent's belief distribution at any time t. The KL-divergence between $\mathcal{N}_{true}$ and agent's belief distribution at time t is:

$$
\begin{aligned}
D_{KL}(\mathcal{N}_{true}||\mathcal{N}_t) =& \frac{1}{2}(tr(\Sigma_t^{-1}\Sigma_{true})+ \\
(\mu_t - \mu_{true})^T\Sigma_t^{-1}(\mu_t - \mu_{true}) - I + \ln(&\frac{det\Sigma_t}{det\Sigma_{true}}))
\end{aligned} \tag{5.20}
$$

where I is dimension of the distributions. Since not action (control state, U) is involved, in this proof, the superscript V in $\mu_t^{(V)}$ is not shown for brevity.

For full-communication, lets assume the agent has communicated with first $J-1$ agents at time t and communicates with the Jth agent at $t+1$. Communicating with the Jth agent decreases agent's accuracy if $D_{KL}(\mathcal{N}_{true}||\mathcal{N}_{t+1}) > D_{KL}(\mathcal{N}_{true}||\mathcal{N}_t)$. Based on Algorithm 1, at this step only μ_t is updated so $\Sigma_{t+1} = \Sigma_t$. Therefore,

$$
\begin{aligned}
& D_{KL}(\mathcal{N}_{true}||\mathcal{N}_{t+1}) > D_{KL}(\mathcal{N}_{true}||\mathcal{N}_t) \equiv \\
& (\mu_{t+1} - \mu_{true})^T\Sigma_t^{-1}(\mu_{t+1} - \mu_{true}) > \\
& (\mu_t - \mu_{true})^T\Sigma_t^{-1}(\mu_t - \mu_{true})
\end{aligned} \tag{5.21}
$$

Since the covariances are equal, the above inequality can be written as:

$$||\mu_{t+1} - \mu_{true}||_2 > ||\mu_t - \mu_{true}||_2 \quad (5.22)$$

Based on Equation 5.14, $\mu_{t+1} = \mu_t + \eta\alpha$ where η is the learning rate. Hence, Equation 5.23 is derived from Equation 5.22.

$$||\mu_t + \eta\alpha - \mu_{true}||_2 > ||\mu_t - \mu_{true}||_2 \quad (5.23)$$

The above inequality is true when $\alpha > 0$. Based on Equation 5.14, and the fact that the Jth agent is the only possible communication target, $\alpha = -\vec{\epsilon}_{p^{(v)}} + \mathbf{\Theta}_{g_e}^T \vec{\epsilon}_{\varphi^{(e)}} + \mathbf{\Theta}_{g_{A_{j'}}}^T \vec{\epsilon}_{\varphi_J}^{(msg)}$. Hence, communication with A_J decreases current agent's accuracy if $\mathbf{\Theta}_{g_e}^T \vec{\epsilon}_{\varphi^{(e)}} + \mathbf{\Theta}_{g_{A_{j'}}}^T \vec{\epsilon}_{\varphi_J}^{(msg)} > \vec{\epsilon}_{p^{(v)}}$.

In our model, an agent estimates the state of its environment in a *localized* manner (i.e. *it communicates neither with a central/global controller nor with all agents all the time*) which has a number of advantages [CGBM04] and is crucial for large networks where using a centralized/global approach is impractical.

Theorem 2 **Localized approach cannot be more accurate than its global counterpart.**

Proof sketch. If each agent has some unique information, some information is lost in the localized case unless all agents are communicated with; in that case, it becomes equivalent to the global approach. However, if an agent does not have any unique information for a particular task, the localized approach may not communicate with this agent and still be as accurate as the global approach.

Therefore, the goal is to maximize estimation accuracy and communication efficiency by choosing to communicate with the most informative agents.

5.5 Experimental results

The model is evaluated for human activity recognition. The proposed model is used in three experiments: (1) skeleton-based human activity recognition in order to evaluate the model when number of agents is reasonably large. (2) multimodal human activity recognition in order to evaluate the model for heterogenous data which is generated from different modalities, and (3) recognition of FoG in PD patients where data is not segmented for different activities and it is generated from different sensors of the same type.

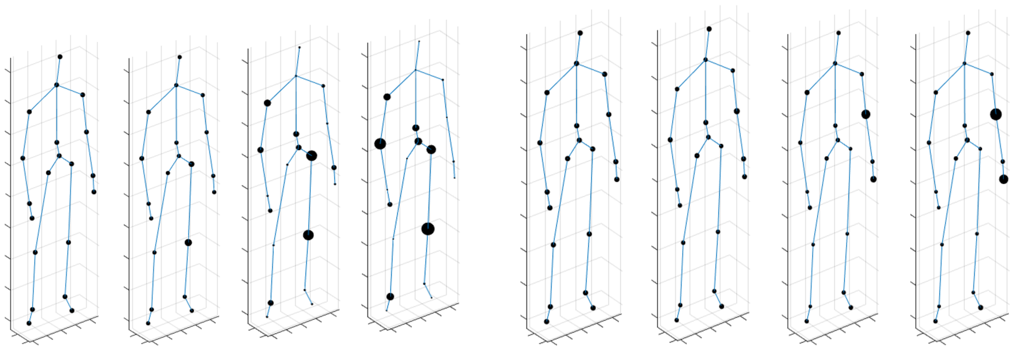
5.5.1 Skeleton-based human activity recognition

Benchmark datasets for activity recognition rarely exceed a few sensors. So the model is evaluated on two benchmark datasets for human activity recognition using Microsoft Kinect skeleton data where each joint is assumed to be monitored by an agent. *KARD* dataset [GRM15] comprises of 18 activities performed by 10 individuals. Each person repeated each activity three times. The dataset includes 540 sequences. The Kinect skeleton has 15 joints, each with three coordinates. *UTD-MHAD* [CJK15b] is a multi-modal dataset which also includes Kinect skeleton data. It has 27 activities performed by eight subjects. Each subject performed each activity four times. After removing three corrupted sequences, the dataset includes 861 sequences. The Kinect skeleton has 20 3-D joints.

Each joint in the skeleton is monitored by a predictive coding agent. For example, the head joint is an agent observing only its 3-D signals and the communication messages from other joints (agents) upon request. However, it does not have access to the observations of the other joints and also other agents' internal models. In order to compare with baselines, the "new person" setup, as in [GRM15] is used where data of one subject is reserved for testing while the model is trained on data of other subjects. First, a dictionary of 50 features is learned from the training set. Inference starts with the head (primary) agent (the joint representing the head of the person) though this does not have to be the case. From the index of the best matched feature for each of the three coordinates and their corresponding optimal shifts, the posterior probability distribution over all possible states (activity categories) is inferred by the primary agent, independently. The agent iteratively refines the belief using the steps shown in Algorithm 2. Communication stops if the change in VFE is less than ϵ $(= 10^{-3})$. The internal model of the primary agent is updated based on the final inference.

Figure 5.1 shows the learned policies for a particular subject for two activity classes. The learned policy for different activities are different. The head agent relies on the agents located in parts of the body with more variations in the environmental signals during that activity. Figure 5.2 shows the final learned policy for a situation where the head agent fails to distinguish between two activities: *Lunge* and *Bowling*. The head agent inferred *Lunge* as *Bowling* half of the times. A sample frame of a subject's posture for each of these activities are shown. The largest circle belongs to the wrist agent (hand agent is not visible in the figure due to its small size). Based on information theory, it is expected that the head agent chooses the agents in the most salient parts of the body during a particular activity (i.e. the signals with less mutual information) [RN16]. Saliency of an agent is measured by the KL-divergence between its belief distribution and that of the head (primary) agent's.

Figure 5.2(b) compares the saliency of different agents' beliefs. A circle's radius is proportional to KL-divergence between distributions. However, this saliency is with respect to the head (primary) agent at the initial step without considering the pairwise similarity between the beliefs of other agents. Two agents might convey the same information so that once the head agent communicates with one of them, the other one is no longer



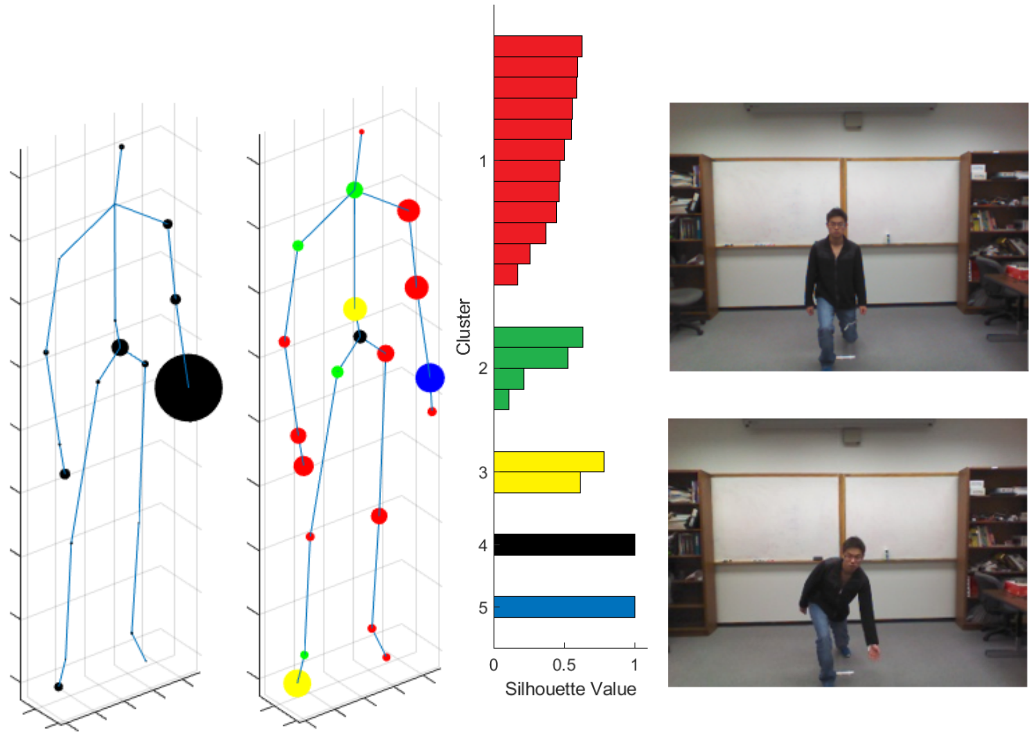
(a) Walking (b) Waving

Fig. 5.1: The learned policies for two activity classes. Number of training iterations (from left to right): 1, 100, 500, 1000. Length of a circle's radius is proportional to the probability of communicating with the corresponding joint-agent.

salient. A non-myopic approach takes the conditional saliency into account. It can be seen that only a subset of the most salient joints are in the learned policy. To visualize this, we grouped the agents' beliefs using k-means clustering and plotted the joints in the same cluster with the same color. The number of clusters is decided based on average number of times the agents communicated for this activity class. The silhouette coefficients indicate the clusters are reasonably compact and homogeneous (ref. Figure 5.2(c)). Even though the saliency of the hip-center agent is less than some of the others, in the policy distribution it has a higher weight because it is alone in its cluster and no other agent's belief is similar to its. Among the more salient joints, at least one from each cluster is present in the learned policy.

Figure 5.3 shows how the optimal policy will be updated over time if a subject performs an activity in an unusual way or some of the agents fail to provide meaningful information. We allowed four agents, located in left shoulder, elbow, wrist and hand, to generate random beliefs. The updated optimal policy for *walking* is shown. The prior policy is the last one shown in Figure 5.1 top row.

Figure 5.4 shows an example of sequential decision-making by an agent for whom to communicate with. It shows how the head agent decides on a sequence of actions to



(a) (b) (c) (d)

Fig. 5.2: (a) Policy when desired state is *Lunge* but the head agent infers *Bowling* from its environmental observations. (b) Saliency of each joint (colors show clusters). (c) Silhouette coefficient. (d) A sample frame from each activity.

decrease the uncertainty. We have intentionally chosen an activity regarding which the head agent is highly uncertain and ends up communicating with six other agents before reaching the final decision. The activity is *Knocking*. First, the head agent infers it as *Jogging*. Refer to the first top left subfigure in Figure 5.4 and the corresponding belief. This belief has high entropy, so the agent communicates with the wrist agent to reduce uncertainty. It can be seen that the maximum belief is changed to the 21st activity which is *Pick up and Throw* (note that throwing involves wrist movement similar to knocking). The communication continues by requesting belief from the hip agent. It reduces the uncertainty in belief by decreasing the second maximum probability. That is, by asking the hip agent, the agent recognizes the activity is not *Lunging*. Finally, the agent reaches

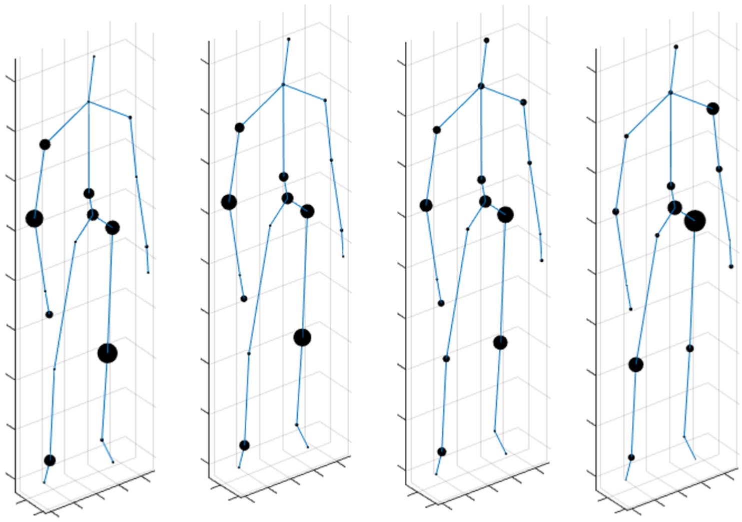


Fig. 5.3: Adapting the prior optimal policy for a special situation where four of the agents generate random beliefs. Data from first subject of UTD-MHAD during walking.

the correct state by communicating with shoulder agent and becomes more certain by communicating with elbow and shoulder center agents.

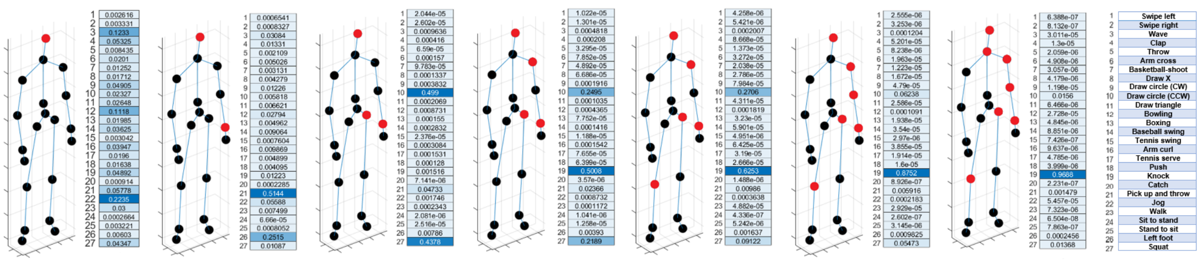


Fig. 5.4: Sequential decision-making for *with whom to communicate*. Red circle denotes the agent $A_{j'}$ selected for communication. Primary agent A_j's belief vector (probability of each environmental state or activity) after communication is shown.

For quantitative evaluation, two cases are considered: (1) the probability of each agent sending random responses is non-zero, and (2) a fixed set of agents, drawn from a uniform distribution, generate random beliefs for a number of trials. We compare our model with two widely-used decision-making methods: (1) an information theoretic technique, Value of Information (ref. Chapter 16 of [RN16]), as a myopic and offline decision-making, and (2) fusion where the posterior probability is computed at a central node as weighted mean of all agents' beliefs. Results are shown in Figure 5.5 and Figure 5.6. When agents randomly fail to provide informative messages, online non-myopic decision-making helps to maintain accuracy by increasing the number of communications. However, when the same agents fail to send informative messages for a long time, updating the agents' models helps the primary agent to adapt its policy; the increase in number of communications is less compared to a non-adaptive approach.

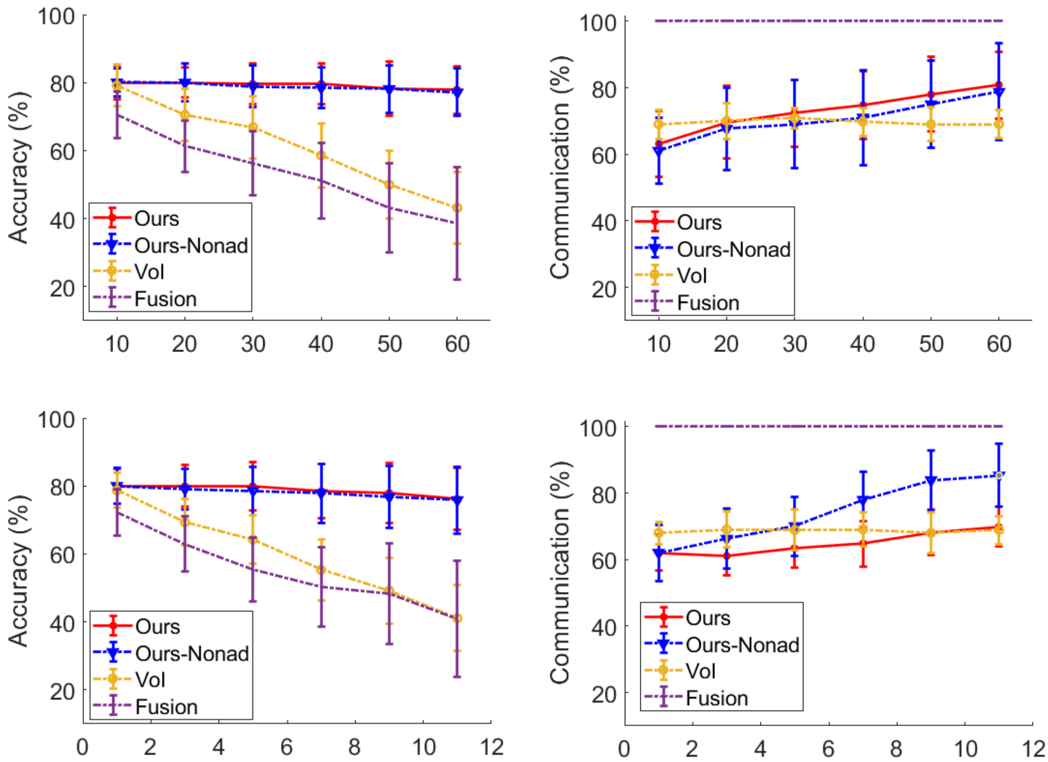


Fig. 5.5: Advantages of online, non-myopic decision-making, as well as online updating of agents' model are shown in these figures. The results are from UTD-MHAD dataset. The plots in the top row show the accuracy and number of communications when each agent has a probability of failure at each point of time. The two plots in the bottom show the same metrics but a fixed number of agents, sampled from a uniform distribution, change their behavior and send random messages for a long time. Nonad and VoI stand for Non-adaptive and Value of Information (a myopic and offline planning method) methods, respectively.

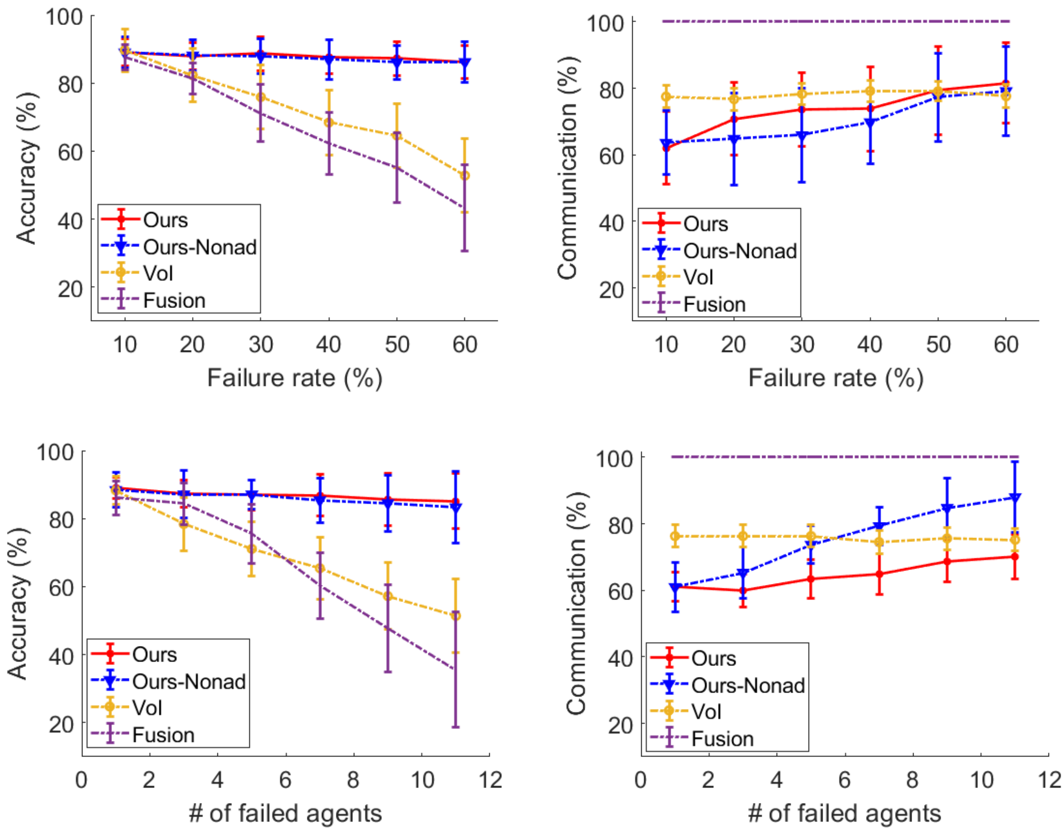


Fig. 5.6: Advantages of online, non-myopic decision-making, as well as online updating of agents' model are shown in these figures. The results are from KARD dataset. The plots in the top row show the accuracy and number of communications when each agent has a probability of failure at each point of time. The two plots in the bottom show the same metrics but a fixed number of agents, sampled from a uniform distribution, change their behavior and send random messages for a long time. Nonad and VoI stand for Non-adaptive and Value of Information (a myopic and offline planning method) methods, respectively.

Table 5.2 shows the head agent's accuracy in recognizing activity classes using different communication protocols. Using optimal policy, the accuracy of recognition is increased. The head agent communicated 63.54% of the time for KARD and 61.32% of the time for UTD-MHAD dataset which is a significant saving in time and resources. Accuracies from references are provided as a baseline. Note that the accuracy of our model also depends on the nature of the chosen generative function, number of parameters, and dimension of hidden state vector. Accuracy can be improved by replacing our linear generative functions with a more sophisticated one.

Table 5.2: Recognition accuracy(%) for the two datasets. "No Comm" and "Full Comm" refer to accuracy of the head agent alone and when the head agent communicates with *all* other agents. "Policy" refers to the case when the head agent uses its optimal policy for communication. "Ref." provides baseline accuracy for the new person setup in [GRM15] and [CJK16] for KARD and UTD (Kinect alone), respectively.

	No Comm	Full Comm	Policy	Baseline
KARD	24.2±1	88.1±1	90.2±3	84.6
UTD-MHAD	18.6±2	73.1±4	80.1±4	74.7

5.5.2 Multimodal human activity recognition

The proposed model is evaluated for multimodal activity recognition on UTD-MHAD dataset which is introduced in the last section where only Kinect skeleton was used. In this section, data from different modalities, namely, depth, skeleton and inertia are used where each sensor modality is assumed to be monitored by an agent. The frame size in depth data is reduced by a factor of 10 to enhance depth agent's efficiency. The agents' generative functions are learned using data from four subjects (subjects 1 through 4) which are excluded from rest of the experiments. These subjects are considered in [CJK15b] as training set, so using them for training allows appropriate comparison.

The inertial (primary) agent starts the communication process since it has the least number of variables (three variables leading to a 6-D feature vector) which incurs lower computational cost. After an independent inference, it communicates with an agent based on the optimal policy and decides to further communicate until the convergence criterion is satisfied. Recognition accuracy for different kinds of communication are shown in the bottom three rows of Table 5.3. Results show the benefit of communication. However, full communication does not guarantee highest accuracy.

Our model is compared with existing methods that have used the same cross-subject setup for training. The results show that even though our model has significantly fewer parameters, communicating using a learned policy yields higher accuracy than most of these models (see Table 5.3). ConvNets [HLWL16] and JTM ConvNets are slightly (1.86% and 2.79%, respectively) more accurate than our model but they have in the order of 60 million parameters. The number of learnable parameters in our model is 67,050. The inertial agent communicated for 301 and 129 of the testing samples with skeleton and depth respectively, but only three times with both.

Table 5.3: Comparison of proposed and existing methods for recognizing 27 actions in the UTD-MHAD dataset.

Method	Accuracy %
ELC-KSVD [ZLZ+14]	76.19
Kinect&Inertial [CJK15b]	79.10
Cov3DJ[HTGES13]	85.58
ConvNets[HLWL16]	86.97
Dawar and Kehtarnavaz[DK18]	86.3
JTMConvNets[WLLH18]	87.90
Our model	**85.11**
No Comm	29.2
Full Comm	84.6

5.5.3 Recognition of gait freeze

The proposed model is also used for recognizing FoG by continuously monitoring PD patients on Dephnet dataset which contains data of 10 subjects. The goal of collecting this dataset was to develop a wearable assistant for PD patients with FoG. FoG is a sudden and transient inability to move involving about 50% of PD patients [BPR+09]. Three acceleration sensors (each has three variables) were used at ankle, knee and hip of the patients. Each sensor is controlled by an agent. The model is used to recognize whether the state is FoG or not; other activities can be stand, walk or turn. Data of the first three patients are used to initialize the generative functions and are excluded from rest of the experiments. A moving average filter with window length 10 is used to smooth the estimations and reduce the false alarms due to noises.

Performance of the model in terms of standard evaluation metrics is shown in Table 5.4. Our results are better than the user-independent results in [BPR+09] (sensitivity of 73.1 and a specificity of 81.6%). This shows the performance of our localized model is comparable to that of centralized/global approaches due to communicating and adapting over time. Second row of Table 5.4 shows the results for the case of inference without communication. In this experiment, the three sensors have the same type but are located in different locations of the body. Therefore, the three agents have different observation, internal models and hence inference accuracy. Without communication, the agent recognized FoG as no event 48% of the times which leads to a low specificity. Communication resolved this issue significantly. Hence, communicating with agents monitoring different locations of the body improves recognition of FoG. Individualized adaption is useful for this application because patients have issues in unique parts of their bodies, and also PD is a progressive disease.

Table 5.4: Experimental results on Daphnet freezing of gait dataset, shown with and without communication.

	Acc.	Sens.	Spec.	Prec.	F-measure
With	89.12	87.15	91.46	92.39	89.69
Without	65.95	82.89	48.26	62.59	71.73

5.6 Summary

We propose an agent model for efficiently predicting its environmental state via selective communication with other agents. The agent is modeled in the predictive coding framework. It learns a communication policy as a mapping from its belief state to *with whom to communicate* in an online and unsupervised manner, without any reinforcement. The proposed model is evaluated for activity recognition from multimodal, multisource and heterogeneous sensor data. The accuracy is comparable to the state-of-the-art even though our model uses significantly fewer parameters and infers the state in a localized manner. The learned policy reduces number of communications and enhances tolerance to communication failures. To the best of our knowledge, this is the first work on learning communication policies by an agent for predicting the state of its environment.

Chapter 6

What to communicate

In the previous chapters, the problems of *when to communicate with whom* have been addressed. *Whenever* the primary agent is not confident about state of the shared environment, it requests the most informative agent to send its belief. In this chapter, using an example, we show that the agents can also decide *what to communicate*, by minimizing same objective function, VFE. Transferring knowledge of activity recognition across sensor networks is selected as a testbed for evaluating the model. The agents ought to recognize daily activities of individuals living in different homes. Each home has different layout, devices and sensors. Behavioral patterns of individuals also vary. Acquiring labeled data for such tasks is costly so transferring knowledge between sensor networks has been proposed in the literature. In this chapter, we show that the predictive coding agents who monitor homes with different layouts can transfer their knowledge efficiently by selecting most informative messages (i.e. deciding what to communicate). The model is evaluated using two publicly available datasets collected from an apartment and a house with different layouts and different wireless sensors. The results show that the agents successfully learn the patterns of daily activities for an individual and communicate using a set of vocabularies. The model is more accurate than existing work in activity recognition using transfer learning and the communication messages are interpretable.

6.1 Introduction

Recognizing activities of daily living (ADL) is a core component of a variety of applications, such as health monitoring, home automation and automatic security surveillance. Researchers have investigated the benefits of transfer learning in making the activity recognition systems more robust and versatile [CFK13]. In this chapter, we show that how predictive coding agents monitoring environments with different physical settings, can actively transfer their knowledge by communication. The agents learn *what to communicate* in different situations so that the necessary information for estimating state of the environment is exchanged.

Similar to previous chapters, the agents are modeled in the predictive coding framework. However, the generative model of agents is different. We use a conditional hierarchical temporal memory (CHTM), a variation of hierarchical temporal memory (HTM) [Geo08] as a mapping between states and observations. HTM is a biologically plausible probabilistic graphical model which can handle both spatial and temporal patterns [Mal11]. Temporal patterns in HTM allows interpreting the sequence of sub-patterns required for activities and also facilitates learning variants of these sequences which lead to the same activity. The predictive coding agent's action is selecting a word (or set of words) from a vocabulary which reduces the uncertainty more.

In this work, it is assumed that a predictive coding agent is trained using a dataset to predict and recognize daily activity patterns of an individual. This agent is called *expert agent* and using a conscious communication, it helps other predictive coding agents monitoring other homes to learn a mapping between sensory patterns and activity labels. The datasets are publicly available and have been introduced in [VKEK10]. Since communication is costly and extra information can make activity recognition even more difficult, active communication helps other agent not only in learning activity patterns but also in learning a policy for sampling most informative data. The task is challenging since the houses have different layouts and various type of sensors. Hence, the agents' sensory observations are not similar and cannot be communicated. A set of vocabularies is used for communication. Experimental results show that our model is more accurate than existing work. Also, patterns and communication messages are interpretable.

6.2 Related work

Transferring trained classifiers from a 'source' to a 'target' dataset is challenging for activity recognition [RG16] since feature spaces of the two datasets, individuals' activity patterns and physical network settings in various environments are completely different [RG18]. However, transferring knowledge is crucial due to lack of sufficient amount of labeled training data. Transfer learning has been successfully applied for identically distributed (i.i.d.) data [RNK06] but the measurements for activity recognition are time-series. Existing work in transfer learning for activity recognition models are categorized to [Bar18] instance-transfer, feature-representation transfer, parameter-transfer. Instance-transfer techniques transfer the source data to the target which is not feasible when the observation spaces are different (i.e. sensors are different). Feature representation transfer approaches [RC11] map activities and sensors from source domain to target domain based on their similarity. An alternative approach is parameter-transfer [vKEK+08] where source and target domain data is mapped into a matrix of meta features, then a Hidden Markov Model has been trained for detecting activities which is more efficient comparing with the previous approaches. Our proposed model is closer to parameter-transfer approach, in the sense that our vocabulary can be considered as a set of meta-features. However, active communication allows the agents to choose the most informative meta-features for each activity and update their internal models accordingly. This helps to efficiency of the model as all meta-features may not be equally informative for an activity or an individual. For example, information about time of the day may help to recognize whether the person is preparing breakfast or dinner. However, information about location is more helpful to recognize whether the person is sleeping in the bedroom or watching TV in living room. The agents should ask for the most informative meta-features using the words in the vocabulary.

6.3 Definitions

In this section, we introduce the terms and concepts relevant to chapter 6 which has not been used in the previous chapters.

Definition 1. (Transfer learning) [vKEK+08] Transfer learning which is also called learning to learn, knowledge transfer, and meta-learning [RG18] refers to techniques that learn a classifier model for a task by incorporating training data from different, but related classification tasks. Source and target are distinguished as tasks that provide us with training data and the task which is the actual classification task, respectively.

Definition 2. (Hierarchical temporal memory (HTM)) [Geo08] is a probabilistic graphical model which consists of nodes in different level of hierarchy. An HTM node is abstracted using a coincidence detector and a mixture of Markov chains to handle both spatial and temporal patterns. An HTM is learned in an unsupervised fashion.

HTM does not conflict but is different with traditional probabilistic models [Fer11] such as Bayesian networks [Pea14], Hierarchical hidden Markov models [FST98], Boltzmann machine [HS+86], and Hellhmholtz machine [DHNZ95]. Bayesian networks utilizes acyclic graphs for topological organization and impose independence assumptions about probability distributions. While Hierarchical hidden Markov architecture model time in a similar way to HTM, they don't consider a hierarchical topology in space. Also, HTM allows incorporating action and attention [Fer11]. Boltzmann and Hellhmholtz machines do not use the temporal structure of data and do not incorporate any assumptions about hierarchy [Fer11].

We propose CHTM as the generative model of our predictive coding agents which differs from traditional HTM in two cases: 1) since our goal is proposing a model for interpretable activity recognition, we allow the agents to condition on labels. The agent monitoring the target dataset does not have access to the correct labels but it learns a mapping between labels and sensory patterns through communication with the expert agent in terms of meta-features (words in the vocabulary), and 2) as opposed to traditional HTM, we allows the nodes in the same level of hierarchy to have different structure. This is important since the observation space of our agents are different. In the other words, sensory patterns of one agent is not observable for the other agent. Hence our agents' generative model contain private nodes (sensory observation of each agent) and shared nodes (words in the vocabulary). Communication is modeled as activating the shared nodes which resemble requesting or sending the value of a particular meta-feature and allows the agent to communicate using a common language.

6.4 Models and methods

In this section we provide the details of our model for decision making about what to communicate using an example of transferring knowledge between agents who monitor different sensor networks. Similar to previous chapters, we use independent predictive coding agents for estimating state of an environment. However, the environment in this application is not shared between agents but has similarities. The agents are supposed to develop a language based on similarities between the environments so that knowledge of state estimation is transferred from one agent to the other. This requires our agents to have a more sophisticated generative model to learn invariant relationships among features and meta-features, and activity classes. In the following we introduce the agent's environment, the agent's model of environment and the communication mechanism.

6.4.1 Agents' environment for activity recognition from sensor networks

Each agent monitors a house to perform activity recognition. It observes the environment using a set of sensors generating binary values. The assumption is that the house layouts and hence sensor configuration is different for different agents. Therefore, an agent already trained using sufficient amount of labeled data (expert agent) will transfer its knowledge to a target agent for whom there is little or no training data available. Differences in sensor networks make the communication challenging since the agents cannot communicate in terms of their sensory observation. However, there are meta-features [vKEK+08] which is common between agents. Although, these meta-features do not contain equal discriminative ability to distinguish between activities (multiple sensors are mapped to the same meta-feature), they provide a common feature space for the agents to communicate with. Taking inspiration from [vKEK+08], in this work, we assume meta-features to be location of sensor and its functionality, as well as time of the day. Time of the day is not a function of sensor space but refers to the time in general. Time intervals are discretized into hours (one through 24) with the step of one. Table 6.1 shows an example of sensors being represented by two sets of meta-features.

Table 6.1: Representation of sensors using meta-features

Sensors	Locations					Functionalities				
	Bedroom	Kitchen	Toilet	Bathroom	Outside	Entrance	Heating	Storage	Clean	Rest
Microwave	0	1	0	0	0	0	1	0	0	0
Stove	0	1	0	0	0	0	1	0	0	0
⋮										
Mattress pressure	1	0	0	0	0	0	0	0	0	1

6.4.2 Agents' model of environment

A predictive coding agent needs a generative model of its environment. In this chapter, a CHTM is used as the generative model of agent from its environment. Similar to HTM, CHTM consists of a coincidence detector and a mixture of Markov chains. However, the temporal grouping is conditioned on the activity classes. Also, the node structures in the same levels of hierarchy can vary.

Similar to temporal grouping is done using Agglomerative Hierarchical Clustering (AHC) [Geo08]. The distance measure for AHC is inverse of transition probability between coincidence patterns.

The agent infers the activity class in two steps. First, it detects the closest coincidence pattern to its observation and then the temporal group for the coincidence. The final belief about activity class is inferred using the inferred temporal group.

$$p(\Psi_i|\Phi_i) = \sum_{g_i} p(\Psi_i|\Phi_i)p(g_i|\Phi_i) \tag{6.1}$$

where Ψ_i is the ith environmental state, Φ_i is the ith observation and g_i is the ith temporal group. It should be noted that temporal groups contain the transition probabilities $p(c_t|c_{t-1})$ where c is the coincidence. A collection of activated coincidence patterns from

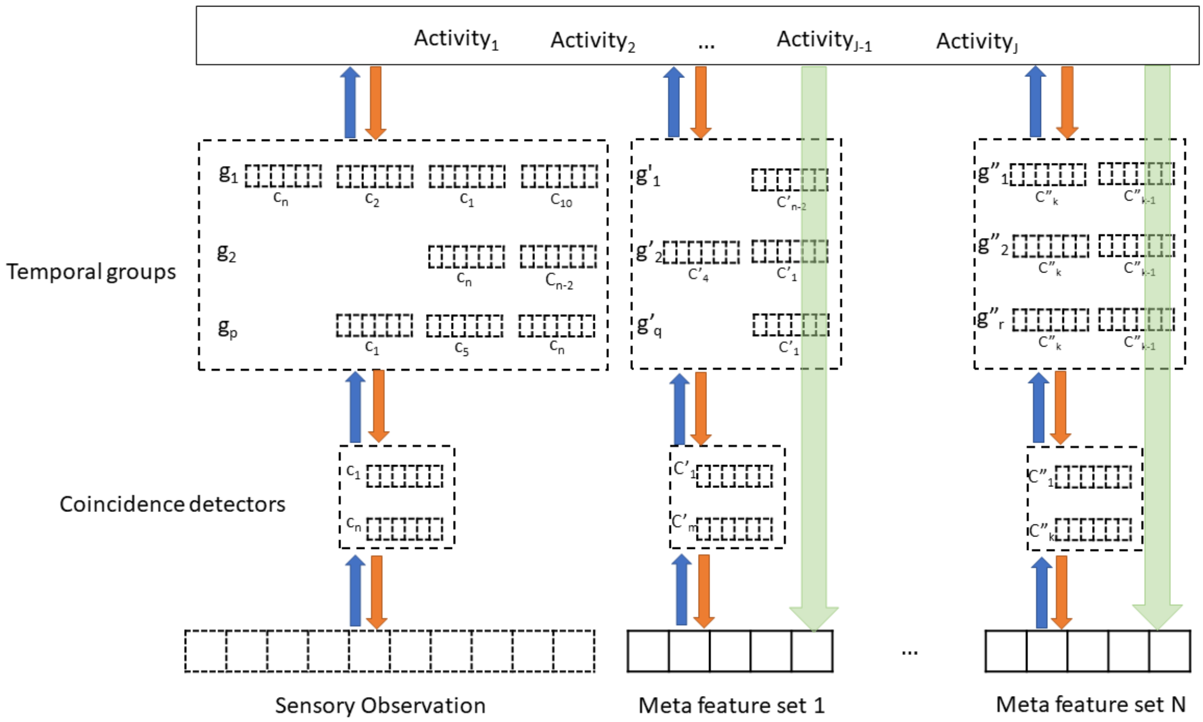


Fig. 6.1: Generative model's architecture for each agent. The dashed blocks are not observable to the other agents. The blue, orange and green arrows show the information flow for inference, feedback and action. Meta-features and activity labels are the common vocabulary between agents through which the agents can communicate.

different nodes produce Φ. Action is selected to reduce uncertainty. In other words, it should maximize $p(\Psi_i, \Phi_i)$. Our predictive coding agents follow the same mechanism as chapter 5 but the index to the most informative node in the hierarchy of Figure 6.1 need to be selected as the optimal action. The target agent who is not confident about inference, send a message containing [*posterior* & *the value of node selected policy*]. Expert agent calculates a posterior using its model using the received value of the node. If the inferred class is different from target agent's inference, the expert agent sends its posterior along with a request for value of another node which can reduce uncertainty most based on policy of expert agent. The target agent updates its internal model at the end of each communication period.

6.5 Experimental results

In this section, we evaluate the model for transferring knowledge of activity recognition between two predictive coding agents (expert agent and target agent).

6.5.1 Data

House B and C from Kasteren dataset [VKEK10] are used to in this chapter. An overview of datasets are shown in Table .

Table 6.2: Information about the datasets [VKEK10] used in this chapter.

	House B	House C
Age	28	57
Gender	Male	Male
Setting	Apartment	House
# of Rooms	2	6
Duration	13 days	18 days
# of Sensors	23	21

List of activities and percentage of each activity in the datasets are provided in Table The set of sensors is not the same in these datasets and the house layouts are different. More information about data can be found in [VKEK10].

6.5.2 Experimental setup

House C is used to train the generative model of expert agent and its policy. The target agent is assumed to monitor House B. Houses B and C are chosen due to including more than 20 sensors. Sensor data is discretized in timeslices of length $\Delta t = 30$ seconds. The expert agent's knowledge need to be transferred to the target agent for which there is limited labeled data. Only, labels from first day of House B has been used for training. In

Table 6.3: List of activities and percentage of participation for each activity for the two datasets used in this chapter.

	House B	House C
Leave house	50.6 %	45.7%
Toileting	0.6%	1.0%
Take shower	0.6%	0.8%
Brush teeth	0.2%	0.4%
Go to bed	30.7%	29.2%
Prepare breakfast	0.5%	0.6%
Prepare dinner	0.2%	1.1%
Get drink	0.2%	0.1%
Other	16.4%	21.1%

our implementation a temporal group has been learned for each activity class. The CHTM has four nodes including, sensory patterns, time of the day (one through 24), locations and functionality of sensors (see Table 6.1). Maximum length of temporal groups for the exper agent in each node is selected using 10 fold cross-validation.

6.5.3 Performance evaluation

We first show the effectiveness of CHTM in learning model of environment and interpretability of the model. Table 6.8 shows accuracy, sensitivity, specificity, precision and f-measure of CHTM for activity recognition using sensor data and meta-features separately. The results show that the discriminative power of meta-features is less than sensory patterns. However, meta-features are the agents' only tools for communications.

Tables 6.5, 6.6 and 6.7 show the temporal groups of each activity class for sensory patterns, location patterns and functionality patterns, respectively. The temporal groups helps to interpret the behavioral patterns of the person who lives in house C. It should be noted that time of the day is itself a temporal information with less resolution comparing to the 30 seconds of sampling period. So transition probability of going to the same pattern

Table 6.4: Evaluation of CHTM in learning model of environment using combination of all features, sensory patterns and meta-features, separately.

	Accuracy	Sensitivity	Specificity	Precision	F-measure
Combination of patterns	97.83	98.82	96.44	97.49	98.15
Sensory patterns	96.38	98.23	93.75	95.72	96.96
Location patterns	74.32	71.87	78.39	84.63	77.73
Time of the day	65.91	86.08	43.30	62.99	72.75
Functionality of sensors	68.81	90.22	52.50	59.13	71.44

is very large. However, some meaningful patterns are found using time of the day. For example, the difference between the time for preparing breakfast and dinner.

We use data of first day from House B for transferring the knowledge from the expert agent (learned from data of House C) via communication. Figure 6.2 shows the part of generative model of target agent (sensory coincidence patterns), after first day of training. It is worth noting that the sensor spaces are quite different and the knowledge is transferred by communication in terms of location, time and functionality of sensors. Figure 6.3 shows the distribution of data collected from House B, in the first day.

The model is then evaluated for recognizing daily activities in House B by the target sensor.

The maximum F-measure for House B is around 60 % using HMM [VKEK10] as generative model of environment.

Figure 6.4 shows the learned policy for what to communicate. It shows probability of exchanging each set of meta-features for each activity class.

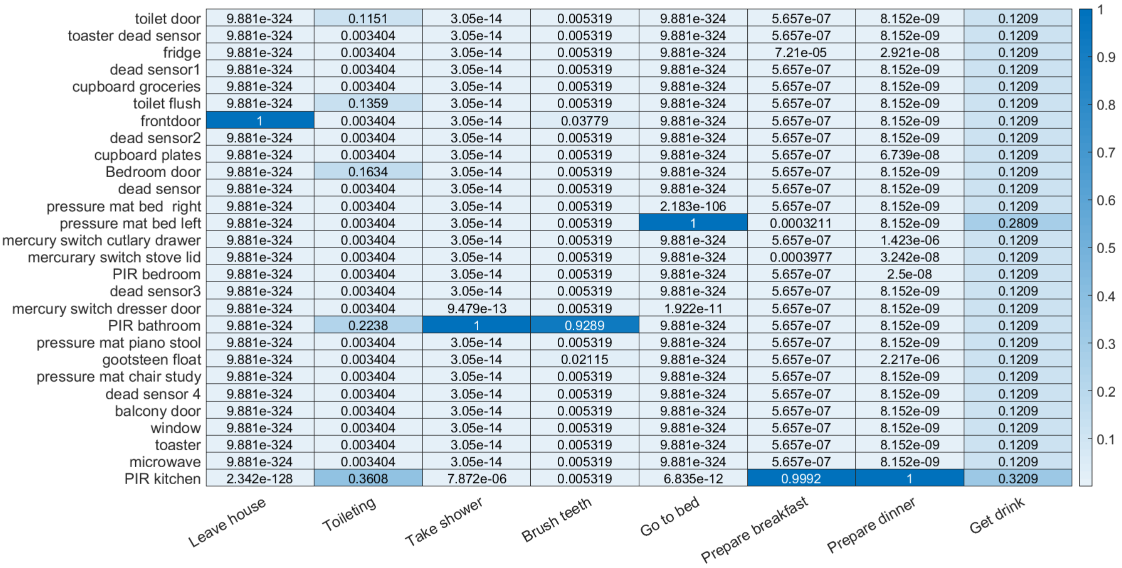


Fig. 6.2: The transferred knowledge about sensory coincidence patterns to the target agent using data of first day. The heatmap shows probability of the sensors firing for each activity lass.

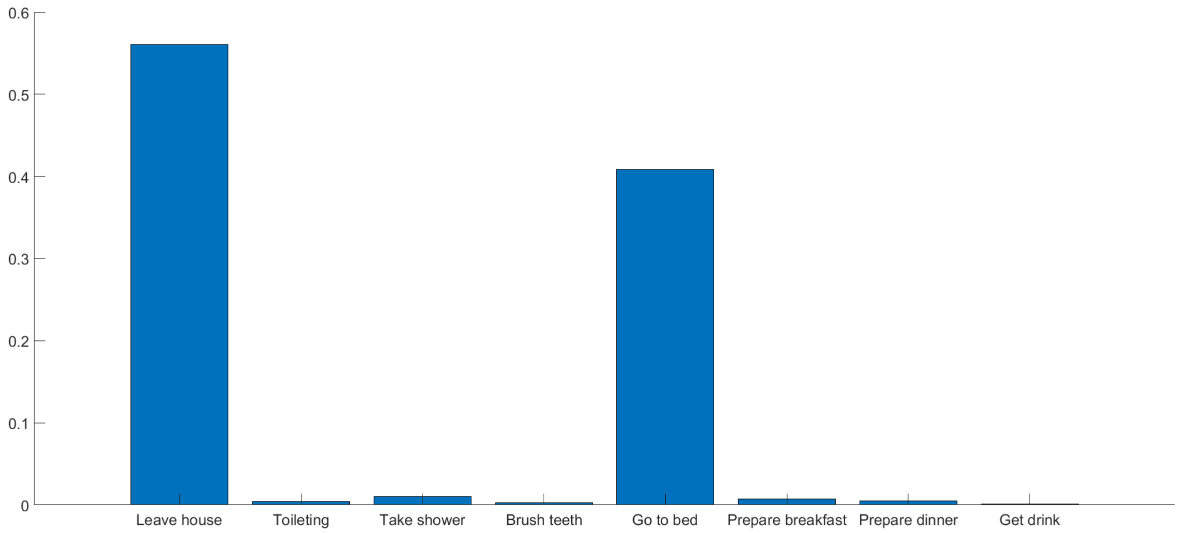


Fig. 6.3: Distribution of data collected from House B, in the first day.

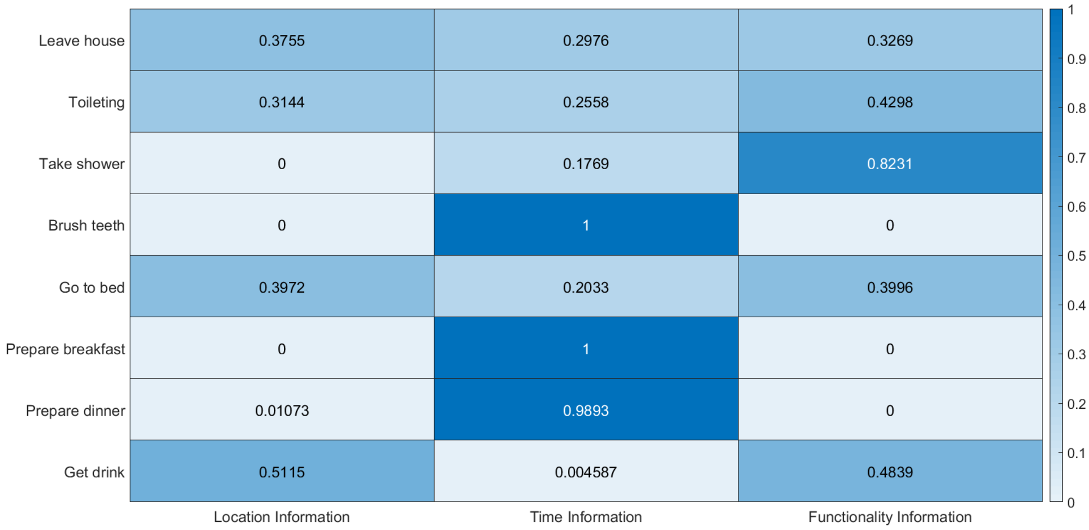


Fig. 6.4: The learned policy for *what to communicate*.

6.6 Summary

Using an example, we showed that the agents can learn a communication policy with respect to *what to communicate* by minimizing the same objective function (VFE). The knowledge from one predictive coding agent is transferred to the other agent by selective communication. The agents communicate using a vocabulary which includes meta-features about sensors attached to different devices in a home setting. Our results show that target agent can successfully sample from expert agent's knowledge and learn a generative model of its environment. The model produces accurate results and is interpretable with respect to individuals' daily activity patterns.

Table 6.5: Temporal groups for sensory patterns. Each group shows a set of activities which are likely to occur together. Groups with one pattern indicate the sensor is fired for a long time.

	Sensors temporal groups
Leave house	g1: bedroom door, cutlery drawer, g2: toilet flush, toilet door, couch pressure mat g3: frontdoor
Toileting	g1: bed right pressure mat, toilet flush upstairs, bathroom swingdoor left, g2: couch, bedroom door, toilet door, g3: toilet flush downstairs
Take shower	g1: couch, toilet flush upstairs, g2: bedroom door, bathroom swing door right, g3: bathtub
Brush teeth	g1: bed right pressure mat, g2: toilet door downstairs, toilet flush downstairs, g3: bathtub, g4: bathroom swingdoor left
Go to bed	g1: toilet flush upstairs, bedroom swingdoor right, g2: bed left pressure mat
Prepare breakfast	g1: bedroom door, cutlery drawer mercury switch, g2: fridge reed, g3: freezer, cupboard bowl and cups, couch pressure mat, g4: microwave reed
Prepare dinner	g1: freezer, reed, cutlery drawer, drawer with keys to backdoor g2: fridge reed, cupboard pots and pans reed, microwave reed, cupboard herbs and plates reed
Get drink	g1: cupboard bowl and cups, fridge reed, g2: couch

Table 6.6: Temporal groups for location patterns.

	location temporal groups
Leave house	g1: outside
Toileting	g1: bathroom
Take shower	g1: bathroom
Brush teeth	g1: bathroom
Go to bed	g1: bedroom
Prepare breakfast	g1: bedroom, kitchen
Prepare dinner	g1: kitchen g2: outside, toilet
Get drink	g1: kitchen g2: bedroom, toilet

Table 6.7: Temporal groups for functionality of device patterns.

	functionality temporal groups
Leave house	g1: entrance
Toileting	g1: clean g2: rest, entrance
Take shower	g1: clean g2: rest, entrance
Brush teeth	g1: storage, rest, clean g2: entrance
Go to bed	g1: rest
Prepare breakfast	g1: storage g2: heat
Prepare dinner	g1: storage, heat g2: rest
Get drink	g1: entrance g2: storage

Table 6.8: Evaluation of transferred knowledge for recognizing daily activities in House B.

	Accuracy	Sensitivity	Specificity	Precision	F-measure
House B	77.52	91.19	53.83	77.40	83.73

Chapter 7

Conclusions

This dissertation presents an agent model that actively and selectively communicates with other agents to predict the state of its environment efficiently. Communication is a challenge when the internal models of other agents is unknown and unobservable. The proposed agent learns communication policies as mappings from its belief state to when, with whom and what to communicate. The same objective function (VFE) is used for perception, action and learning. The proposed agent model is evaluated for Predicting the state of an agent's partially-observable environment from multimodal, multisource and heterogeneous sensor data. Human activity recognition is chosen as a testbed due to its importance in healthcare services and secure surveillance. In our model each sensor or network of sensors is monitored by an agent who rationally decide to communicate with other agents monitoring other sensors. The recognition accuracy on benchmark datasets is comparable to the state-of-the-art, even though our model has significantly fewer parameters and infers the state in a localized manner. The learned policy reduces number of communications. The agent is tolerant to communication failures and can recognize the reliability of each agent from its communication messages. To the best of our knowledge, this is the first work on learning communication policies by an agent for predicting the state of its environment. The proposed model can be used for state estimation and prediction in a dynamic Internet of Things environment where using a predefined communication protocol is not feasible.

Bibliography

[ADWW98] Mark R Adler, Alvah B Davis, Robert Weihmayer, and Ralph W Worrest. Conflict-resolution strategies for nonhierarchical distributed agents. In *Distributed Artificial Intelligence*, pages 139–161. Elsevier, 1998. 19

[AS18] Stefano V Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018. 4

[ASLN+14] Nikolay Atanasov, Bharath Sankaran, Jerome Le Ny, George J Pappas, and Kostas Daniilidis. Nonmyopic view planning for active object classification and pose estimation. *IEEE Transactions on Robotics*, 30(5):1078–1090, 2014. 3

[BAM18] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 26

[Bar18] Paulo Miguel Santos Barbosa. Human activities recognition: a transfer learning approach. 2018. 79

[BAT18] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, 2018. 1, 4, 47

[BCP+18] Graeme Best, Oliver M Cliff, Timothy Patten, Ramgopal R Mettu, and Robert Fitch. Dec-mcts: Decentralized planning for multi-robot active perception. *The International Journal of Robotics Research*, page 0278364918755924, 2018. 3

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 30, 56

[BDN+16] Farshid Hassani Bijarbooneh, Wei Du, Edith C-H Ngai, Xiaoming Fu, and Jiangchuan Liu. Cloud-assisted data fusion and sensor selection for internet of things. *IEEE Internet of Things Journal*, 3(3):257–268, 2016. 3

[Bel52] Richard Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*, 38(8):716–719, 1952. 50

[BGHT+14] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. mhealthdroid: a novel framework for agile development of mobile health applications. In *International Workshop on Ambient Assisted Living*, pages 91–98. Springer, 2014. 27, 36

[BKMS17] Christopher L Buckley, Chang Sub Kim, Simon McGregor, and Anil K Seth. The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*, 2017. viii, 7, 13, 15, 17, 55, 56

[BM09] Nicholas J Butko and Javier R Movellan. Optimal scanning for faster object detection. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*, pages 2751–2758. IEEE, 2009. 3

[BMTH11] Majid Bahrepour, Nirvana Meratnia, Zahra Taghikhaki, and Paul JM Havinga. Sensor fusion-based activity recognition for parkinson patients. In *Sensor Fusion-Foundation and Applications*. InTech, 2011. 28

[Bog15] Rafal Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 2015. 17, 20

[Bog17] Rafal Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211, 2017. 7, 35, 58, 62

[BPR+09] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and Gerhard Troster. Wearable assistant for parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):436–446, 2009. 27, 40, 74, 75

[CFK13] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and information systems*, 36(3):537–556, 2013. 78

[CGBM04] Timothy H Chung, Vijay Gupta, Joel W Burdick, and Richard M Murray. On a decentralized active sensing strategy using mobile sensor platforms in a network. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 2, pages 1914–1919. IEEE, 2004. 65

[CHL13] Yi-Lei Chen, Chiou-Ting Candy Hsu, and Hong-Yuan Mark Liao. Simultaneous tensor decomposition and completion using factor priors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (1):1, 2013. 34

[CJK15a] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Improving human action recognition using fusion of depth camera and inertial sensors. *IEEE Transactions on Human-Machine Systems*, 45(1):51–61, 2015. 28

[CJK15b] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 168–172. IEEE, 2015. 27, 36, 37, 40, 41, 42, 56, 66, 73, 74

[CJK16] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. A real-time human action recognition system using depth and inertial sensor fusion. *IEEE Sensors Journal*, 16(3):773–781, 2016. xi, 73

[CJK17] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools and Applications*, 76(3):4405–4425, 2017. 26, 27, 29

[CM11] Daniel Chandler and Rod Munday. *A dictionary of media and communication*. OUP Oxford, 2011. 12

[CS10] Federico S Cattivelli and Ali H Sayed. Diffusion strategies for distributed kalman filtering and smoothing. *IEEE Transactions on automatic control*, 55(9):2069–2084, 2010. 2, 3

[CTCT18] Alok Chowdhury, Dian Tjondronegoro, Vinod Chandran, and Stewart Trost. Physical activity recognition using posterior-adapted class-based fusion of multiaccelerometer data. *IEEE journal of biomedical and health informatics*, 22(3):678–685, 2018. 39, 41

[CTPK15] Cagatay Catal, Selin Tufekci, Elif Pirmit, and Guner Kocabag. On the use of ensemble of classifiers for accelerometer-based activity recognition. *Applied Soft Computing*, 37:1018–1022, 2015. 41

[DB02] Joachim Denzler and Christopher M Brown. Information theoretic sensor data selection for active object recognition and state estimation. *IEEE Trans. on pattern analysis and machine intelligence*, 24(2):145–157, 2002. 3, 51

[DGR+18] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Michael Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. *arXiv preprint arXiv:1810.11187*, 2018. 4, 48, 49

[DHNZ95] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995. 80

[DHWG19] Derui Ding, Qing-Long Han, Zidong Wang, and Xiaohua Ge. A survey on model-based distributed control and filtering for industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 15(5):2483–2499, 2019. 3

[DK18] Neha Dawar and Nasser Kehtarnavaz. Real-time continuous detection and recognition of subject-specific smart tv gestures via fusion of depth and inertial sensing. *IEEE Access*, 6:7019–7028, 2018. 28, 40, 74

[DP04] Sophie Deneve and Alexandre Pouget. Bayesian multisensory integration and cross-modal spatial links. *Journal of Physiology-Paris*, 98(1-3):249–258, 2004. 19

[DPDJ12] Ezequiel Alejandro Di Paolo and Hanne De Jaegher. The interactive brain hypothesis. *Frontiers in human neuroscience*, 6:163, 2012. 12

[DZB+14] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in neural information processing systems*, pages 1269–1277, 2014. 34

[EAG17] Sergio Escalera, Vassilis Athitsos, and Isabelle Guyon. Challenges in multi-modal gesture recognition. In *Gesture Recognition*, pages 1–60. Springer, 2017. 26

[EGZ09] Robert Eidenberger, Thilo Grundmann, and Raoul Zoellner. Probabilistic action planning for active scene modeling in continuous high-dimensional domains. In *2009 IEEE International Conference on Robotics and Automation*, pages 2412–2417. IEEE, 2009. 3

[EK99] Jamie S Evans and Vikram Krishnamurthy. Hidden markov model state estimation with randomly delayed observations. *IEEE Transactions on Signal Processing*, 47(8):2157–2166, 1999. 2

[FDK09] Karl J Friston, Jean Daunizeau, and Stefan J Kiebel. Reinforcement learning or active inference? *PloS one*, 4(7):e6421, 2009. 4, 8, 13, 20, 48, 51, 54

[Fer11] David Rozado Fernández. *Analysis and extension of hierarchical temporal memory for multivariable time series*. PhD thesis, Universidad Autónoma de Madrid, 2011. 80, 81

[FF15] Karl J Friston and Christopher D Frith. Active inference, communication and hermeneutics. *Cortex*, 68:129–143, 2015. 13, 18, 60

[FR12] Charles W Fox and Stephen J Roberts. A tutorial on variational bayesian inference. *Artificial intelligence review*, 38(2):85–95, 2012. 53

[Fri05] Karl Friston. A theory of cortical responses. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1456):815–836, 2005. 32, 57, 62

[Fri10] Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127, 2010. 7, 8, 9, 13, 47, 50, 55, 62

[FSM12] Karl Friston, Spyridon Samothrakis, and Read Montague. Active inference and agency: optimal control without cost functions. *Biological cybernetics*, 106(8-9):523–541, 2012. 9, 51, 54, 55, 60

[FST98] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998. 80

[GAGF17] Raffaele Gravina, Parastoo Alinia, Hassan Ghasemzadeh, and Giancarlo Fortino. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Information Fusion*, 35:68–80, 2017. 27, 28, 44

[Geo08] Dileep George. *How the brain might work: A hierarchical and temporal model for learning and recognition*. Stanford University Palo Alto, California, 2008. 78, 80, 83

[GHRA17] Markus Grotz, Timothée Habra, Renaud Ronsse, and Tamim Asfour. Autonomous view selection and gaze stabilization for humanoid robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1427–1434. IEEE, 2017. 3

[GMH13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013. 2

[GRM15] Salvatore Gaglio, Giuseppe Lo Re, and Marco Morana. Human activity recognition process using 3-d posture data. *IEEE Trans. Human-Machine Systems*, 45(5):586–597, 2015. xi, 66, 73

[GSS93] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993. 2

[HC16] Sojeong Ha and Seungjin Choi. Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 381–388. IEEE, 2016. 41

[HKB20] Masoumeh Heidari Kapourchali and Bonny Bannerjee. EPOC: Efficient perception via optimal communication. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020. 6

[HLWL16] Yonghong Hou, Zhaoyang Li, Pichao Wang, and Wanqing Li. Skeleton optical spectra based action recognition using convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016. 39, 40, 74

[Hos17] Yedid Hoshen. Vain: Attentional multi-agent predictive modeling. In *NIPS*, pages 2701–2711, 2017. 4, 47, 48

[HS+86] Geoffrey E Hinton, Terrence J Sejnowski, et al. Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(282-317):2, 1986. 80

[HTGES13] Mohamed E Hussein, Marwan Torki, Mohammad Abdelaziz Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*, volume 13, pages 2466–2472, 2013. 40, 74

[HWZ+18] Kuan Han, Haiguang Wen, Yizhen Zhang, Di Fu, Eugenio Culurciello, and Zhongming Liu. Deep predictive coding network with local recurrent processing for object recognition. In *NIPS*, pages 9201–9213, 2018. 7

[HYZW18] Xue Han, Hongping Yan, Junge Zhang, and Lingfeng Wang. Acm: Learning dynamic multi-agent cooperation via attentional communication model. In *Int. Conf. on Artificial Neural Networks*, pages 219–229. Springer, 2018. 4, 47, 48

[JG18] Dinesh Jayaraman and Kristen Grauman. End-to-end policy learning for active visual categorization. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 3

[JM08] Craig J Johns and Jan Mandel. A two-stage ensemble kalman filter for smooth data assimilation. *Environmental and Ecological Statistics*, 15(1):101–110, 2008. 2

[JNSS18] Artur Jordao, Antonio Nazare, Jessica Sena, and William Schwartz. Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art. *arXiv preprint arXiv:1806.05226*, 2018. 39

[JPC07] Shihao Ji, Ronald Parr, and Lawrence Carin. Nonmyopic multiaspect sensing with partially observable markov decision processes. *IEEE Transactions on Signal Processing*, 55(6):2720–2730, 2007. 3

[JY15] Wenchao Jiang and Zhaozheng Yin. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the*

23rd ACM international conference on Multimedia, pages 1307–1310. ACM, 2015. 41

[KB18a] Masoumeh Heidari Kapourchali and Bonny Banerjee. Multiple heads outsmart one: A computational model for distributed decision making. In *CogSci 2018*, pages 1779–1784, July 2018. 3, 6

[KB18b] Masoumeh Heidari Kapourchali and Bonny Banerjee. Unsupervised feature learning from time-series data using linear models. *IEEE Internet of Things Journal*, 2018. 6, 30, 56, 57

[KB19] Masoumeh Heidari Kapourchali and Bonny Banerjee. State estimation via communication for monitoring. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2019. 3, 6, 48

[KFF07] James M Kilner, Karl J Friston, and Chris D Frith. Predictive coding: an account of the mirror neuron system. *Cognitive processing*, 8(3):159–166, 2007. 7

[KGO12] Hilbert J Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182, 2012. 8

[KHG11] Veronika Koltunova, Jesse Hoey, and Marek Grześ. Goal-oriented sensor selection for intelligent phones. In *Proceedings of the 2011 international workshop on Situation activity & goal awareness*, pages 83–88. ACM, 2011. 3

[KKHI05] Chris Kreucher, Keith Kastella, and Alfred O Hero Iii. Sensor management using an active sensing approach. *Signal Processing*, 85(3):607–624, 2005. 3

[KMGS14] Leszek Kaliciak, Hans Myrhaug, Ayse Goker, and Dawei Song. On the duality of specific early and late fusion strategies. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–8. IEEE, 2014. 26

[KR96] David C Knill and Whitman Richards. *Perception as Bayesian inference*. Cambridge University Press, 1996. 51

[KRS14] Igor Kviatkovsky, Ehud Rivlin, and Ilan Shimshoni. Online action recognition using covariance of shape and motion. *Computer Vision and Image Understanding*, 129:15–26, 2014. 28

[KTKL14] Adil Khan, Ali Tufail, Asad Khattak, and Teemu Laine. Activity recognition on smartphones via sensor-fusion and kda-based svms. *International Journal of Distributed Sensor Networks*, 10(5):503291, 2014. 28

[LCJK14] Kui Liu, Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Fusion of inertial and depth sensor data for robust hand gesture recognition. *IEEE Sensors Journal*, 14(6):1898–1903, 2014. 27

[LGG+18] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018. 27

[LJRW16] Yue Li, Devesh K Jha, Asok Ray, and Thomas A Wettergren. Sensor selection for passive sensor networks in dynamic environment: A dynamic data-driven approach. In *2016 American Control Conference (ACC)*, pages 4924–4929. IEEE, 2016. 3

[LMHPP13] Nathan Lepora, Uriel Martinez-Hernandez, Giovanni Pezzulo, and Tony Prescott. Active bayesian perception and reinforcement learning. In *Intel-*

ligent Robots and Systems, International Conference on, pages 4735–4740. IEEE, 2013. 44

[LMWY13] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2013. 34

[Mal11] Davide Maltoni. Pattern recognition by hierarchical temporal memory. *Available at SSRN 3076121*, 2011. 78

[MD00] Heinz J Müller and Rose Dieng. *Computational Conflicts: Conflict Modeling for Distributed Intelligent Systems*. Springer Science & Business Media, 2000. 19

[MT08] Manuel Mazo and Paulo Tabuada. On event-triggered and self-triggered control over sensor/actuator networks. In *2008 47th IEEE Conference on Decision and Control*, pages 435–440. IEEE, 2008. 3

[MZ93] Stéphane Mallat and Zhifeng Zhang. Matching pursuit with time-frequency dictionaries. Technical report, Courant Institute of Mathematical Sciences New York United States, 1993. 31, 57

[NGC19] Cameron Nowzari, Eloy Garcia, and Jorge Cortés. Event-triggered communication and control of networked systems for multi-agent consensus. *Automatica*, 105:1–27, 2019. 3

[NM14] Rudy R Negenborn and Jose Maria Maestre. Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, 34(4):87–97, 2014. 3

[NS12] Sebastian Nowozin and Jamie Shotton. Action points: A representation for low-latency online human action recognition. *Microsoft Research Cambridge, Tech. Rep. MSR-TR-2012-68*, 2012. 28

[Ole99] Daniel E Oleary. Identifying conflicting probabilistic knowledge in multiple knowledge base systems. In *Agents' conflicts-Papers from the AAAI workshop-Technical report WS-99-08*, pages 7–13, 1999. 19

[OS05] Reza Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184. IEEE, 2005. 2, 3

[PCBH17] Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125, 2017. 26, 28

[PCT+10] Marc Ponsen, Tom Croonenborghs, Karl Tuyls, Jan Ramon, Kurt Driessens, Jaap Van den Herik, and Eric Postma. Learning with whom to communicate using relational reinforcement learning. In *Interactive Collaborative Information Systems*, pages 45–63. Springer, 2010. 35, 48

[Pea14] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014. 80

[Pre98] Lutz Prechelt. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761–767, 1998. 44

[PYW+17] Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017. 49

[QAT+15] Guo-Jun Qi, Charu Aggarwal, Deepak Turaga, Daby Sow, and Phil Anno. State-driven dynamic sensor selection and prediction with state-stacked sparseness. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 945–954. ACM, 2015. 3

[RB99] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79, 1999. 7, 8

[RC11] Parisa Rashidi and Diane J Cook. Activity knowledge transfer in smart environments. *Pervasive and Mobile Computing*, 7(3):331–343, 2011. 79

[RDML05] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. Activity recognition from accelerometer data. In *Aaai*, volume 5, pages 1541–1546, 2005. 28

[RG16] Seyed Ali Rokni and Hassan Ghasemzadeh. Plug-n-learn: automatic learning of computational algorithms in human-centered internet-of-things applications. In *Proceedings of the 53rd Annual Design Automation Conference*, page 139. ACM, 2016. 79

[RG18] Seyed Ali Rokni and Hassan Ghasemzadeh. Autonomous training of activity recognition algorithms in mobile sensors: A transfer learning approach in context-invariant views. *IEEE Transactions on Mobile Computing*, 17(8):1764–1777, 2018. 79, 80

[RN16] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016. 7, 8, 10, 49, 50, 67, 70

[RNK06] Rajat Raina, Andrew Y Ng, and Daphne Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 713–720. ACM, 2006. 79

[SF+16] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, pages 2244–2252, 2016. 48

[SF17] Maciej Stachura and Eric Frew. Communication-aware information-gathering experiments with an unmanned aircraft system. *Journal of Field Robotics*, 34(4):736–756, 2017. 3

[SKH14] Jadwiga Sobieska-Karpińska and Marcin Hernes. Identification of the knowledge conflicts' sources in the architecture of cognitive agents supporting decision-making process. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1157–1162. IEEE, 2014. 19

[SM12] Carl Scarrott and Anna MacDonald. A review of extreme value threshold es-timation and uncertainty quantification. *REVSTAT–Statistical Journal*, 10(1):33–60, 2012. 44

[Spa08] Matthijs TJ Spaan. Cooperative active perception using pomdps. In *AAAI 2008 workshop on advancements in POMDP solvers*, 2008. 3

[Spr17] Michael W Spratling. A review of predictive coding algorithms. *Brain and cognition*, 112:92–97, 2017. 7

[SWO15] Yash Satsangi, Shimon Whiteson, and Frans A Oliehoek. Exploiting submodular value functions for faster dynamic sensor selection. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 3

[SWOS18] Yash Satsangi, Shimon Whiteson, Frans A Oliehoek, and Matthijs TJ Spaan. Exploiting submodular value functions for scaling up active perception. *Autonomous Robots*, 42(2):209–233, 2018. 3, 49, 50, 52

[Tod07] Emanuel Todorov. Linearly-solvable markov decision problems. In *NIPS*, pages 1369–1376, 2007. 8

[Vas18] Alexander Yurievich Vasilyev. Optimal control of eye-movements during visual search. *IEEE Transactions on Cognitive and Developmental Systems*, 2018. 3

[vKEK+08] Tim van Kasteren, Gwenn Englebienne, Ben JA Kröse, et al. Recognizing activities in multiple contexts using transfer learning. In *AAAI Fall Symposium: AI in Eldercare: New Solutions to Old Problems*, pages 142–149, 2008. 79, 80, 82

[VKEK10] TLM Van Kasteren, Gwenn Englebienne, and Ben JA Krose. Transferring knowledge of activity recognition across sensor networks. In *International Conference on Pervasive Computing*, pages 283–300. Springer, 2010. xi, 78, 85, 87

[VLS18] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1510–1517, 2018. 27

[VR16] Tuomas Välimäki and Risto Ritala. Optimizing gaze direction in a visual navigation task. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1427–1432. IEEE, 2016. 3

[WHJ14] Changyun Wei, Koen V Hindriks, and Catholijn M Jonker. The role of communication in coordination protocols for cooperative robot teams. In *ICAART (2)*, pages 28–39, 2014. 39

[Wic77] Wayne A Wickelgren. Speed-accuracy tradeoff and information processing dynamics. *Acta psychologica*, 41(1):67–85, 1977. 44

[WLLH18] Pichao Wang, Wanqing Li, Chuankun Li, and Yonghong Hou. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowledge-Based Systems*, 158:43–53, 2018. 74

[WLTS06] Jamie A Ward, Paul Lukowicz, Gerhard Troster, and Thad E Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1553–1567, 2006. 28

[WSH03] Danny Weyns, Elke Steegmans, and Tom Holvoet. A model for active perception in situated multi-agent systems. In *Proc. of the 1st European Workshop on MAS*, 2003. 3

[WST14] Kishan Wimalawarne, Masashi Sugiyama, and Ryota Tomioka. Multitask learning meets tensor factorization: task imputation via convex optimization. In *Advances in neural information processing systems*, pages 2825–2833, 2014. 34

[WVDM00] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000. 2

[WYL+18] Zifeng Wang, Zheng Yu, Qing Ling, Dimitris Berberidis, and Georgios B Giannakis. Decentralized rls with data-adaptive censoring for regressions over large-scale networks. *IEEE Transactions on Signal Processing*, 66(6):1634–1648, 2018. 3

[YLL17] Rose Yu, Guangyu Li, and Yan Liu. Tensor regression meets gaussian processes. *arXiv preprint arXiv:1710.11345*, 2017. 34

[YLP+15] Özgür Yürür, Chi Liu, Charith Perera, Min Chen, Xue Liu, and Wilfrido Moreno. Energy-efficient and context-aware smartphone sensor employment. *IEEE Trans. on Vehicular Technology*, 64(9):4230–4244, 2015. 3

[YLW16] Scott Cheng-Hsin Yang, Mate Lengyel, and Daniel M Wolpert. Active sensing in the categorization of visual patterns. *Elife*, 5:e12215, 2016. 3

[YQL16] Yuan Yuan, Lei Qi, and Xiaoqiang Lu. Action recognition by joint learning. *Image and Vision Computing*, 55:77–85, 2016. 27

[YYL03] Yiming Ye, Xun Yi, and Juhnyoung Lee. Performance and attention in multiagent object search team. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 33(2):257–264, 2003. 3

[ZJ05] Yongmian Zhang and Qiang Ji. Sensor selection for active information fusion. In *AAAI*, pages 1229–1234, 2005. 3

[ZJ06] Yongmian Zhang and Qiang Ji. Active and dynamic information fusion for multisensor systems with dynamic bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(2):467–472, 2006. 3

[ZLZ+14] Lijuan Zhou, Wanqing Li, Yuyao Zhang, Philip Ogunbona, Duc Thanh Nguyen, and Hanling Zhang. Discriminative key pose extraction using extended lc-ksvd for action recognition. In *Digital lmage Computing: Techniques and Applications (DICTA), 2014 International Conference on*, pages 1–8. IEEE, 2014. 40, 74

[ZSB+11] Bin Zhang, Chris Sconyers, Carl Byington, Romano Patrick, Marcos E Orchard, and George Vachtsevanos. A probabilistic fault detection approach: Application to bearing fault detection. *IEEE Transactions on Industrial Electronics*, 58(5), 2011. 44

[ZSR02] Feng Zhao, Jaewon Shin, and James Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal processing magazine*, 19(2):61–72, 2002. 3

[ZWZ17] Lei Zou, Zi-Dong Wang, and Dong-Hua Zhou. Event-based control and filtering of networked systems: A survey. *International Journal of Automation and Computing*, 14(3):239–253, 2017. 3

[ZYO+15] Roie Zivan, Harel Yedidsion, Steven Okamoto, Robin Glinton, and Katia Sycara. Distributed constraint optimization for teams of mobile sensing agents. *Autonomous Agents and Multi-Agent Systems*, 29(3):495–536, 2015. 3

Appendix A

Derivation of Eq. 5.14 is provided. The agent's belief is updated to minimize VFE. Based on definition:

$$F = -\ln p(\tilde{\mu}, \tilde{\Phi}) + C = -\ln p(\tilde{\Phi}|\tilde{\mu})p(\tilde{\mu}) + C$$

In the paper, agent's belief after communication is shown with $\vec{\mu}^{(v)}_{t+1}$ since the agent is using the new observation due to a_t. In the following derivation, we use the same time notation $t+1$ to be consistent with the paper, however, it should be noted that $t+1$ refers to the current time and so $\vec{\varphi}^{(msg)}_{t+1}$ has been observed. Therefore, F turns to:

$$F = -\ln p(\vec{\varphi}^{(e)}, \vec{\varphi}^{(msg)}_{1:t+1}|\vec{\mu}^{(v)}_{t+1}, \vec{\mu}^{(u)}_{1:t+1})p(\vec{\mu}^{(v)}_{t+1}, \vec{\mu}^{(u)}_{1:t+1}) + C$$

Considering that $\vec{\varphi}^{(e)}$ is independent from $\vec{\mu}^{(u)}$,

$$F = -\ln\Big(\big(p(\vec{\varphi}^{(e)}|\vec{\mu}^{(v)}_{t+1})\prod_{\tau}^{t+1} p(\vec{\varphi}^{(msg)}_{\tau}|\vec{\mu}^{(v)}_{\tau}, \vec{\mu}^{(u)}_{\tau})\big)\big(p(\vec{\mu}^{(v)}_{t+1})\prod_{\tau}^{t+1} p(\vec{\mu}^{(u)}_{\tau+1}|\vec{\mu}^{(v)}_{\tau}, \vec{\mu}^{(u)}_{\tau})\big)\Big) + C =$$
$$-\ln p(\vec{\varphi}^{(e)}|\vec{\mu}^{(v)}_{t+1}) - \sum_{\tau}^{t+1} \ln p(\vec{\varphi}^{(msg)}_{\tau}|\vec{\mu}^{(v)}_{\tau}, \vec{\mu}^{(u)}_{\tau}) - \ln p(\vec{\mu}^{(v)}_{t+1}) - \sum_{\tau}^{t+1} \ln p(\vec{\mu}^{(u)}_{\tau+1}|\vec{\mu}^{(v)}_{\tau}, \vec{\mu}^{(u)}_{\tau})) + C$$

Plugging the Gaussians in the above equation, F can be written as:

$$F = -\ln\Big(\frac{1}{\sqrt{(2\pi)^{2M}|\mathbf{\Sigma}_{\varphi^{(e)}}|}}exp\big(-\frac{1}{2}(\varphi^{(e)} - g_e(\vec{\mu}^{(v)}_{t+1}, \mathbf{\Theta}_{g_e}))^T$$
$$\mathbf{\Sigma}^{(-1)}_{\varphi^{(e)}}(\varphi^{(e)} - g_e(\vec{\mu}^{(v)}_{t+1}, \mathbf{\Theta}_{g_e})))\Big) - \sum_{\tau}^{t+1}\ln\Big(\frac{1}{\sqrt{(2\pi)^{I}|\mathbf{\Sigma}_{\varphi^{(msg)}_{\tau}}|}}$$
$$exp\big(-\frac{1}{2}(\varphi^{(msg)}_{\tau} - g_{A_{j'}}(\vec{\mu}^{(v)}_{\tau}, \vec{\mu}^{(u)}_{\tau}, \mathbf{\Theta}_{g_{A_{j'}}})^T\mathbf{\Sigma}^{-1}_{\varphi^{(msg_{j'})}}(\varphi^{(msg)}_{\tau}$$
$$- g_{A_{j'}}(\vec{\mu}^{(v)}_{\tau}, \vec{\mu}^{(u)}_{\tau}, \mathbf{\Theta}_{g_{A_{j'}}}))\Big) - \ln\Big(\frac{1}{\sqrt{(2\pi)^{I}|\mathbf{\Sigma}_{p^{(e)}}|}}exp\big(-\frac{1}{2}$$
$$(\vec{\mu}^{(v)}_{t+1} - \vec{v}_p)^T\mathbf{\Sigma}^{(-1)}_{p^{(e)}}(\vec{\mu}^{(v)}_{t+1} - \vec{v}_p))\Big) - \sum_{\tau}^{t+1}\ln\Big(\frac{1}{\sqrt{(2\pi)^{J}|\mathbf{\Sigma}_{\pi}|}}exp\big($$
$$(\vec{\mu}^{(u)}_{\tau+1} - g_\pi(\vec{\mu}^{(u)}_{\tau}, \vec{\mu}^{(v)}_{\tau}, \mathbf{\Theta}_{\pi}))^T\mathbf{\Sigma}^{-1}_{\pi}(\vec{\mu}^{(u)}_{\tau+1} - g_\pi(\vec{\mu}^{(u)}_{\tau}, \vec{\mu}^{(v)}_{\tau}, \mathbf{\Theta}_{\pi})))\Big) + C$$

Using the logarithm rules, above equation becomes:

$$F = -\frac{1}{2}\Big(- \ln|\mathbf{\Sigma}_{\varphi^{(e)}}| - (\varphi^{(e)} - g_e(\vec{\mu}^{(v)}_{t+1}, \mathbf{\Theta}_{g_e}))^T\mathbf{\Sigma}^{(-1)}_{\varphi^{(e)}}(\varphi^{(e)}$$
$$- g_e(\vec{\mu}^{(v)}_{t+1}, \mathbf{\Theta}_{g_e})) - \sum_{\tau}^{t+1}\big(\ln|\mathbf{\Sigma}_{\varphi^{(msg)}_{\tau}}| - (\varphi^{(msg)}_{\tau} - g_{A_{j'}}(\vec{\mu}^{(v)}_{\tau},$$
$$\vec{\mu}^{(u)}_{\tau}, \mathbf{\Theta}_{g_{A_{j'}}})^T\mathbf{\Sigma}^{-1}_{\varphi^{(msg_{j'})}}(\varphi^{(msg)}_{\tau} - g_{A_{j'}}(\vec{\mu}^{(v)}_{\tau}, \vec{\mu}^{(u)}_{\tau}, \mathbf{\Theta}_{g_{A_{j'}}}))-$$
$$\ln|\mathbf{\Sigma}_{p^{(e)}}| - (\vec{\mu}^{(v)}_{t+1} - \vec{v}_p)^T\mathbf{\Sigma}^{(-1)}_{p^{(e)}}(\vec{\mu}^{(v)}_{t+1} - \vec{v}_p)) - \sum_{\tau}^{t+1}\big(\ln|\mathbf{\Sigma}_{\pi}|-$$
$$g_\pi(\vec{\mu}^{(u)}_{\tau}, \vec{\mu}^{(v)}_{\tau}, \mathbf{\Theta}_{\pi}))^T\mathbf{\Sigma}^{-1}_{\pi}(\vec{\mu}^{(u)}_{\tau+1} - g_\pi(\vec{\mu}^{(u)}_{\tau}, \vec{\mu}^{(v)}_{\tau}, \mathbf{\Theta}_{\pi})))\Big) + C$$

where derivative of F with respect to $\vec{\mu}^{(v)}_{t+1}$, is:

$$\frac{\partial F}{\partial \vec{\mu}^{(v)}_{t+1}} = \frac{\partial g_e(\vec{\mu}^{(v)}_{t+1}, \mathbf{\Theta}_{g_e})^T}{\partial \vec{\mu}^{(v)}_{t+1}} \mathbf{\Sigma}^{-1}_{\varphi^{(e)}} (\vec{\varphi}^{(e)} - g_e(\vec{\mu}^{(v)}_{t+1}, \mathbf{\Theta}_{g_e})) + \sum_{\tau=1}^{t+1}$$
$$\frac{\partial g_{A_{j'}}(\vec{\mu}^{(v)}_\tau, \vec{\mu}^{(u)}_\tau, \mathbf{\Theta}_{g_{A_{j'}}})^T}{\partial \vec{\mu}^{(v)}_\tau} \mathbf{\Sigma}^{-1}_{\varphi^{(msg_{j'})}} (\vec{\varphi}^{(msg_{j'})}_\tau - g_{A_{j'}}(\vec{\mu}^{(v)}_\tau, \vec{\mu}^{(u)}_\tau,$$
$$\mathbf{\Theta}_{g_{A_{j'}}})) - \mathbf{\Sigma}^{-1}_{p^{(e)}} (\vec{\mu}^{(v)}_{t+1} - \vec{v}_p) + \sum_{\tau=1}^{t+1} \frac{\partial g_\pi(\vec{\mu}^{(u)}_\tau, \vec{\mu}^{(v)}_\tau, \mathbf{\Theta}_\pi)^T}{\partial \vec{\mu}^{(v)}_\tau} \mathbf{\Sigma}^{-1}_\pi$$
$$(\vec{\mu}^{(u)}_\tau - g_\pi(\vec{\mu}^{(u)}_\tau, \vec{\mu}^{(v)}_\tau, \mathbf{\Theta}_\pi))$$

where $\vec{\epsilon}_{\varphi^{(e)}} = \mathbf{\Sigma}^{-1}_{\varphi^{(e)}} (\vec{\varphi}^{(e)} - g_e(\vec{\mu}^{(v)}, \mathbf{\Theta}_{g_e})))$, $\vec{\epsilon}_{\varphi^{(msg)}} = \mathbf{\Sigma}^{-1}_{\varphi^{(msg_{j'})}} (\vec{\varphi}^{(msg_{j'})} - g_{A_{j'}}(\vec{\mu}^{(v)}, \vec{\mu}^{(u)}, \mathbf{\Theta}_{g_{A_{j'}}}))$, $\vec{\epsilon}_{p^{(v)}} = \mathbf{\Sigma}^{-1}_{p^{(e)}} (\vec{\mu}^{(v)} - \vec{v}_p)$ and $\vec{\epsilon}_\pi = \mathbf{\Sigma}^{-1}_\pi (\vec{\mu}^{(u)} - g_\pi(\vec{\mu}^{(u)}, \vec{\mu}^{(v)}, \mathbf{\Theta}_\pi)$.

Therefore,

$$\frac{\partial F}{\partial \vec{\mu}^{(v)}_{t+1}} = \frac{\partial g_e(\vec{\mu}^{(v)}_{t+1}, \mathbf{\Theta}_{g_e})^T}{\partial \vec{\mu}^{(v)}_{t+1}} \vec{\epsilon}_{\varphi^{(e)}} +$$
$$\sum_{\tau=1}^{t+1} \frac{\partial g_{A_{j'}}(\vec{\mu}^{(v)}_\tau, \vec{\mu}^{(u)}_\tau, \mathbf{\Theta}_{g_{A_{j'}}})^T}{\partial \vec{\mu}^{(v)}_\tau} \vec{\epsilon}_{\varphi^{(msg)}_\tau} -$$
$$\vec{\epsilon}_{p^{(v)}} + \sum_{\tau=1}^{t+1} \frac{\partial g_\pi(\vec{\mu}^{(u)}_\tau, \vec{\mu}^{(v)}_\tau, \mathbf{\Theta}_\pi)^T}{\partial \vec{\mu}^{(v)}_\tau} \vec{\epsilon}_\pi$$