Observational Learning: Imitation Through an Adaptive Probabilistic Approach

Sheida Nozari[1,2], Lucio Marcenaro[1], David Martin[2] and Carlo Regazzoni[1]
Department of Engineering and Naval architecture (DITEN), University of Genoa, Italy[1]
Intelligent systems lab, University Carlos III de Madrid, Spain[2]
email addresses: sheida.nozari@edu.unige.it
{lucio.marcenaro, carlo.regazzoni}@unige.it, dmgomez@ing.uc3m.es

***Abstract*—This paper proposes an adaptive method to enable imitation learning from expert demonstrations in a multi-agent context. Our work employs the inverse reinforcement learning method to a coupled Dynamic Bayesian Network to facilitate dynamic learning in an interactive system. This method studies the interaction at both discrete and continuous levels by identifying inter-relationships between the objects to facilitates the prediction of an expert agent's demonstrations. We evaluate the learning procedure in the scene of learner agent based on probabilistic reward function. Our goal is to estimate policies that predicted trajectories match the observed one by minimizing the Kullback-Leiber divergence. The reward policies provide a probabilistic dynamic structure to minimize the abnormalities.**

***Index Terms*—imitation learning, multi-learning, Q-learning, Dynamic Bayesian network, performance analysis**

I. Introduction

Imitation learning (IL) [1] approaches aim to mimic an expert behavior by transferring skills through observations and by following the demonstrations step-by-step [2]. However, imitating each step often becomes impracticable when the learning-agent and the environment are different from those in the demonstration. Meanwhile, using IL to track and reach a target in motion is still a challenging task. In many cases, the agent does not have to follow the expert unconditionally. Instead, it must care about the demonstrator's intention or the goal-based imitation [3]. A moving object can be modeled as a series of interactions with its surroundings such that its dynamics result from forces that act on it over time [4].

Modeling and understanding expert demonstrations (e.g., trajectories) are essential tasks in the successes of multi-agent learning in a dynamic environment such as intelligent transportation [5], autonomous systems [6]–[8] and sports tracking data [9]. In order for autonomous multi-agent to learn such skills, they need a supervision signal that indicates the goal of the expected behavior. Typically, this supervision can come from a reward function in reinforcement learning (RL) that specifies which states and actions are desirable [10]. Recent advances in RL have improved IL to learn complicated behaviors in dynamic environments [11]. The integration of both modalities, RL and IL, enables the learning of complex skills from raw sensory observations [12]. However, the reward function in RL is task-specific, and the difficulty of manually specifying a reward function represents a significant barrier to the broader applicability of RL in complex observations [13]. Inverse reinforcement learning (IRL) [14] bypasses this issue by assuming that an agent receives the sequences of observation-action tuples. It tries to learn how to map observations to actions from these sequences through estimating a reward function. By approximating this function rather than directly learning the state-action, the apprentice is able to learn a reward function in new scenarios that explains the observed expert behavior. Moreover, it allows adapting to disarrays in the dynamics of the environment [15].

Accordingly, the demonstrations can be explained by a set of configurations between the moving objects each time instant. Therefore, we can provide complex models that explain the interaction between objects and their surroundings [16]. We aim to take advantage of such interactions in a probabilistic manner through a coupled Dynamic Bayesian Network (DBN) structure. DBNs have been used for representing temporal relationships of the agent and a dynamic target. It is the case of predictive models based on objects' locations and their time derivatives [17]–[20]. To build this interaction model, we first use a set of spatial zones in a scene where the configurations are valid based on multiple expert demonstrations. Then, we use transitions between the zones to track observations by employing a set of Kalman filters [21] coupled with a Particle filter [22] method to take advantage of both discrete and continuous variables under an interaction assumption. Finally, we employ the IRL approach using Q-network [23] to extract the probabilistic reward function regarding the detected abnormalities to match and evaluate the learner agent state trajectory (evidence) with the expert's demonstration (expectation). We employ simulated data to validate the proposed method performance at the interacting rules into probabilistic models.

Our contribution are summarized as follows: *i)* we employ IRL approach to a coupled DBN structure that facilitates the characterization of objects' dynamics and their inter-relationships; *ii)* we learn a probabilistic multiple reward functions without exploiting the expert demonstration explicitly; *iii)* Inferences from the proposed integrated method are used to minimize the abnormalities depending on the state of their surroundings. Learning a probabilistic reward function allows us to take uncertainty about the agent's dynamic into account, which reduces learning bias due to model errors.

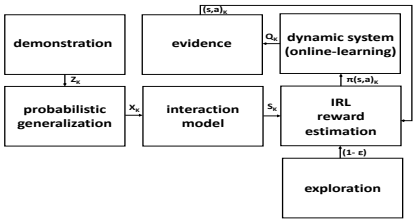


Fig. 1: Overview of the system.

II. Trajectory Representation

Probabilistic Graphical Models (PGMs) employ graph-based representation to encoding a variety of multi-dimensional random variables and represent causal relationships among them [24]. A particular type of PGM is the Dynamic Bayesian Network (DBN) [25]. Due to its hierarchical nature, DBN can express the temporal relationship between high-level variables (capturing abstract semantic information of the world) and low-level distributions (capturing rough sensory information of the environment) with their respective evolution through time. Recent works studied several algorithms for inference in PGMs based on a data-driven way [26], [27]. A modern inference mechanism, namely, the Markov Jump Particle Filter (MJPF) presented in [26] can be employed to facilitate the generation of behavior based on DBN models learned computationally from data.

A. Dynamic Interaction Model

Let Z_k^1 and Z_k^2 be the observed positions of two entities, namely *Teacher (Te)* and *Reference Target (T_{Ref})*. Both agents are assumed to interact with each other at a given time instant k. Let us consider a KF which uses zero order motion dynamical equation:

$$\tilde{X}_k = A\tilde{X}_{k-1} + w_k, \tag{1}$$

where $\tilde{X}_k$ represents the object's state composed of its generalized coordinate positions and their velocities in a time instant k, such that $\tilde{X}_k = [\boldsymbol{x}\ \dot{\boldsymbol{x}}]^\intercal$ where $\boldsymbol{x} \in \mathbb{R}^d$ and $\dot{\boldsymbol{x}} \in \mathbb{R}^d$. d represents the number of coordinates of the environment. In (1), $A = [A_1\ A_2]$ is a dynamic model matrix where $A_1 = [I_d\ 0_{d,d}]^\intercal$ and $A_2 = 0_{2d,d}$. I_n represents a square identity matrix of size n and $0_{l,m}$ is a $l \times m$ null matrix. w_k represents the prediction noise which is here assumed to be zero-mean Gaussian for all variables in X_k with a covariance matrix Q, such that $w_k \sim \mathcal{N}(0, Q)$. The proposed model in (1) suggests that moving objects will rest in a quasi-static location and only random noise perturbations, modeled by w_k will affect their states. At each time instant k a new measurement Z_k is made and it is assumed a linear relationship between Z_k and $\tilde{X}_k$, such that:

$$Z_k = H\tilde{X}_k + v_k, \tag{2}$$

where $H = [I_d\ 0_{d,d}]$ is the observation matrix that maps hidden states ($\tilde{X}_k$) to measurement (Z_k) and v_k is the measurement noise which is assumed to be zero-mean Gaussian with covariance R, such that, $v_k \sim \mathcal{N}(0, R)$. The deviations from predicted velocities are approximated using: $\dot{\boldsymbol{x}} = H^{-1}\big(Z_t - H\tilde{X}_{k-1}\big)$. A joint state space vector (System generalized states) is defined as $\tilde{\boldsymbol{X}}_k$ and consists of both *Te* and T_{Ref} states at each time instant k, such that:

$$\tilde{\boldsymbol{X}}_k = [\tilde{X}_k^1\ \ \tilde{X}_k^2]^\intercal, \tag{3}$$

where $\tilde{X}_k^1$ and $\tilde{X}_k^2$ represent the Generalized States (GS) of Te and T_{Ref} respectively. To learn a situation model for our system, we perform a coupled DBN [16] by using two vocabularies, Te and T_{Ref}. The vocabularies are based on generalized joint states coming from training examples that describe a specific type of interaction between the objects. Each vocabulary is composed of configurations where $\tilde{\boldsymbol{X}}_k$ data is clustered. Each configuration represents a region where quasi-linear models are valid to present the interactive dynamical system over time (Fig.2.a). Vocabularies are defined as:

$$\boldsymbol{S}^i = \{s_1^i, s_2^i, \ldots, s_{L_i}^i\}, \tag{4}$$

where L_i is the total number of prototypes associated with the *object* i and s_l^i indexes the cluster of generalized joint states that favors *object* i's motion.

In a time instant k, each *object* i is represented by a situation state $S_k^i \in \boldsymbol{S}^i$. Active situation state from different objects are considered together as an activated configuration. For our case, the activated configuration at the time instant k is written as $D_k = [S_k^T, S_k^{T_{Ref}}]^\intercal$. Consequently, it is possible to the define a dictionary containing possible configuration, such that:

$$\boldsymbol{D} = \{D^1, D^2, \cdots, D^M\}, \tag{5}$$

where D^m encodes a given identified configuration, M represents the total number of configurations (situation states combinations) and $D_k \in \boldsymbol{D}$. So, the configurations are created based on the different situation states related to the considered objects at the same time instant. Thus, $\boldsymbol{D}$ defines the whole system's discretization and the corresponding dynamics.

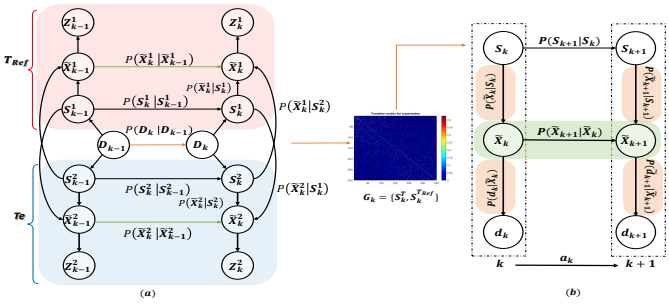


Fig. 2: a) A coupled-DBN for the inteaction between Te and T_{Ref}. b) The provided learning model by L.

Transition models at the discrete level. By observing the configurations over time, it is possible to estimate a set of temporal transition matrices that encode the probabilities of passing from a current configuration to another one to estimate $p(D_k|D_{k-1}, t_k)$, where t_k encodes the time spent in the current word D_{k-1}.

Linear dynamic model at the continuous level. The object's motion can be modeled based on quasi-constant velocity, that is a function of the previously obtained regions S^i.

B. Probabilistic Learning Model

Each situation configuration S^i includes the *Te* and T_{Ref} features $[(X_{bi}, V_{bi}), (X_{\beta j}, V_{\beta j})]_s$, where X_{bi} and $X_{\beta j}$ represent the position of *Te* and T_{Ref}, and V_{bi} and $V_{\beta j}$ represent the velocity of *Te* and T_{Ref}. Here, we can associate an average distance d_Z to each configuration, that is a difference between X_{bi} and $X_{\beta j}$.

The situation position is not meaningful because the agent in a dynamic environment is usually required to dealing with limited information. In order, by moving from one reference configuration to the other one, the system computes the distance in each time instant. This feature must be comparable with the current model. Current model is based on the Learner agent *(L)* and the current target *(T_{Cur})* in the real-time through the online learning. Also, in the current model, we consider the interaction between *L* and T_{Cur} as a configuration in each time instant $[(X_{bi}, V_{bi}), (X_{\beta j}, V_{\beta j})]_{cur}$. Therefore, the *L* measures the distance from the target which will change each time due to the action performed.

The *L* uses the transition model estimated from the situation model (i.e. by observing the interaction between the teacher and the target) to learn a new DBN encoding the dynamic behaviour followed by the teacher to reach the dynamic target (Fig.2-b). The transition model encodes the probability of moving from a certain configuration $[(X_{bi}, V_{bi}), (X_{\beta j}, V_{\beta j})]_{k-1}$ to another one $[(X_{bi}, V_{bi}), (X_{\beta j}, V_{\beta j})]_k$ in the situation model. In this way, the *L* can predict the expected future configurations based on the dynamic transition rules encoded in the model and imitates a similar trajectory as the one it observed from the *T*. We employ the Markov Jump Particle Filter (MJPF) [28] which uses a combination of Particle Filter (PF) and Kalman Filter (KF) for prediction and inference purposes. Using the MJPF allows to predict the interaction among configurations at different levels: i) at the discrete level, to predict future configurations by means of PF which uses the transition probabilities ($p(D_k|D_{k-1})$) encoded in the transition model as a proposal distribution to propagate a set of particles realizing the predicted discrete variables (i.e. configurations); ii) at the continuous level, where velocity measurements and motion estimation of the states are are predicted using a bank of KFs.

Both levels provide a qualitative comparison between the current model's evidence in real-time and the corresponding prediction of the situation model through the learning procedure. Belief in hidden variables can be updated after receiving a new observation. Here the observations are the estimated distances $d(Te, T_{Ref})$. Then we estimate the expected d_z in the next time instant.

III. LEARNING DYNAMIC MULTIPLE REWARD

The objective of learning the reward policies is to integrate the IL with IRL by taking turns to i) optimize imitation policies that minimize the abnormalities (imitation loss). Hence, here, learning is relatively robust to modeling errors. ii) provide a probabilistic dynamic structure by an interactive reward estimation.

This work hypothesizes that during the learning phase, the learner uses a probabilistic interactively model. It employs the model in a Q-network context for *i)* learning a multiple reward function and *ii)* regulating the learners' movement in the learning phase. We explain both contributions as below.

A. Reward function

Two different policies are considered:

Policy I. Learning to minimize the difference between the current learner's action, a_k and the mean action $\dot{\mathcal{C}}_{k-1}$ of the activated configuration $S_k \in \boldsymbol{\mathcal{S}}_{train}$ in the situation model, such that:

$$P_k^I = d_{\mathcal{M}}\big(g_a(S_k),\ a_k\big), \tag{6}$$

where $g_a(\cdot)$ is a function that extracts the action-distribution from a GS-distribution, such that $g_a(S_k) \sim \mathcal{N}(\dot{\mathcal{C}}_k,\ \Sigma_k^a)$ and Σ_k^a is the action's covariance information. $d_{\mathcal{M}}(\mathtt{X}, \mathtt{x})$ is the Mahalanobis distance between a distribution $\mathtt{X}$ and a point $\mathtt{x}$. $S_k \sim \mathcal{N}(\tilde{\mathcal{C}}_k,\ \Sigma_K)$, which can be written as:

$$S_k = \underset{\mathcal{S}_m}{\operatorname{argmin}} \|s_k - \mathcal{C}_m\|_2. \tag{7}$$

Policy II. Learning to minimize the divergence between the distribution over the learner state (S_k) (calculated after taking an action a_{k-1}) and the discrete probability $p(S_k|S_{k-1})$ from the situation model (calculated by transition model $(d_z)_k$). The term S_{k-1}, required in $p(S_k|S_{k-1})$, is calculated based on Eq.(7).

The PF is employed to provide distributions over the learner state to have dynamic weight in the reward computation. The goal is to track the distributed state sequence (P_k) of a dynamic model. The word distributed emphasizes imperfect measurement from the current model by adapting noise to the learner state. The probability distribution over the learner state allows us to represent the uncertainty about the agent's dynamics. For estimating d, two sources of information are required, the prior knowledge on how the d_K is expected to evolve and a measurement model related to evaluated (P_k). Here, we use the transition model to find the expected d, and we calculate Kullback–Leibler (KL) divergence [29] between two estimation, the d_z and d_{pi} to adjust the learner state. The KL presents a control input on the particles' weight. KL is used to refine the particles by comparing the expectation and the current model measurements. The particles with the more likelihood to the prediction survive, and we use the mean of them to have probabilistic reward by considering the uncertainty. The policy II can be written as:

$$P_k^{II} = d_{\mathcal{M}}\big(g_s(d_{k|k-1}) \mid\mid \bar{X} \sum_{i=1}^{n} d_{pi}\big), \tag{8}$$

where $g_s(\cdot)$ as a function that extracts the state-distribution from a GS-distribution, such that $g_s(S_k) \sim \mathcal{N}(\mathcal{C}_k,\ \Sigma_t^s)$. Σ_t^s is

the state's covariance information.
This paper considers both policies in parallel as a reward:

$$R_k^{II} := P_k^I + P_k^{II}. \tag{9}$$

B. Abnormality measurement

This work proposes an abnormality measurement based on the KL divergence between the situation states $p(\tilde{S}_k^i|\tilde{S}_{k-1}^i)$ and the evidence $p(d_k|\tilde{X}_k)$, such that:

$$\lambda_k^i = \int p(\tilde{S}_k^i|\tilde{S}_{k-1}^i) log \frac{p(\tilde{S}_k^i|\tilde{S}_{k-1}^i)}{p(d_k|\tilde{X}_k)}. \tag{10}$$

The values of λ_k^i indicates how much the prediction is supported by the observation. If the observation matches the prediction, then λ_k^i is close to 0. Otherwise the prediction deviates from the observation which leads to a high value of λ_k^i (close to 1) revealing the presence of an abnormality.

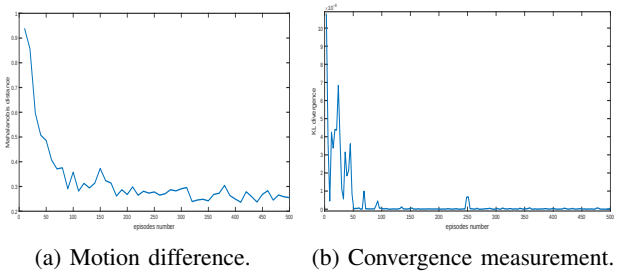
IV. Results

In this section, we provide numerical results to validate the proposed method. We consider a table of trained data where the L, chases T_{Cur} in a 40×40 space. In training data, the L's motion is described by 8 different motion unit-vectors associated with the cardinal and intercardinal directions. The T_{Cur} motions consists in a horizontal dynamics along the x axis at a fixed height point $y_{T_{Cur}}$. Accordingly, the T_{Cur} can move in two senses: right or left inside the interval $[x_{T_{Cur}}^{(min)}, x_{T_{Cur}}^{(max)}]$. The T_{Cur}'s dynamics consists of a continuous motion in one sense until it reaches an interval boundary. Then, it starts moving in the opposite sense covering only the defined interval points. The speed of T_{Cur} movements is different from the T_{Ref} in the situation model to guarantees that the L learns to reach the target in a new scenario. The following parameters are employed for simulation purposes: $y_{T_{Cur}} = 15$, $x_{T_{Cur}}^{(min)} = -15$ and $x_{T_{Cur}}^{(max)} = 15$. Results related to the capabilities of detecting abnormalities and evaluating the current model are explained in detail as follows.

Abnormality detection. Evaluating the current model's configurations during the learning phase is employed to detect abnormalities. Training includes 500 episodes from a different start position that each episode presents 8 trajectories. It means L tries 4000 trajectories through 500 different start positions. Fig. 3-a shows the result of motion's difference between the L and T at the continuous level at time K by using Mahalanobis distance.

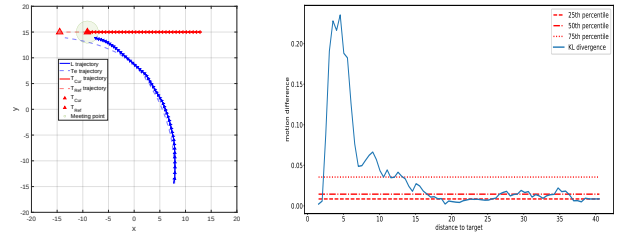
Fig. 3-b shows abnormality estimation in case of the divergence between the current model's configuration and the situation model's prediction at the discrete level at time $K+1$ through KL divergence measurements. From both figures, it is possible to see how high abnormality values are present in the learning's initial portion. Once the L learns the reward policies, the measurements go down dramatically. In Fig. 3-b, although the divergence measurements is not too high (the highest value is 11×10^{-3}), it learns to minimize it.

Current model evaluation. Here the situation model is available as a ground truth. To evaluate the current model's efficiency, we translate the testing phase's result to a switching DBN based on L and T_{Cur} interaction. Fig. 4-b shows the result of a comparison between the motions generated based on the situation model and the respective evidence of the translated current model by using KL measurements.



(a) Motion difference. (b) Convergence measurement.

Fig. 3: Comparison between learned policies and the expert.



(a) Example of matched trajectories. (b) Motion difference based on distance from target.

Fig. 4: Comparison between the situation model and the current mode.

As Fig.4-b shows, when the L's distance to the T_{Cur} is between [15,40], where the L follows the expert trajectory, the abnormality estimation is lower than other positions. In the range [10,15], the measurement increases gradually because the L tries to adapt to the T_{Ref} behavior. The highest difference belongs to the distance between [0, 5] to the target, where the L's motion is goal-based. However, most of the abnormality measurements (75%) are less than 0.03, that as we mentioned previously, values close to 0 indicate that evidence matches with the expectation.

V. Conclusion

In this paper, we proposed an adaptative probabilistic model for IL based on observation. Algorithms for performing inferences and learning the probabilistic reward structure are presented, which enables the learning-agent to take uncertainty appropriately into account. Our method demonstrates learning from an interaction model to estimate the reward function through online learning. Experimental results show the capability to minimize the abnormalities while learning the policies from the demonstrations. Comparisons between the simulated Learner agent and encoded DBN configurations in the proposed model can encode multiple IRL policies. Future works include more complex interactions between objects, such as more than one learner agent, to create robust DBN structures for IRL.

REFERENCES

[1] S. Schaal *et al.*, "Learning from demonstration," *Advances in neural information processing systems*, pp. 1040–1046, 1997.

[2] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 358, no. 1431, pp. 537–547, 2003.

[3] D. Verma and R. P. Rao, "Goal-based imitation as probabilistic inference over graphical models," in *Advances in neural information processing systems*. Citeseer, 2006, pp. 1393–1400.

[4] D. Campo, A. Betancourt, L. Marcenaro, and C. Regazzoni, "Static force field representation of environments based on agents' nonlinear motions," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, pp. 1–15, 2017.

[5] W. Jiang, J. Lian, M. Shen, and L. Zhang, "A multi-period analysis of taxi drivers' behaviors based on gps trajectories," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[6] M. Baydoun, M. Ravanbakhsh, D. Campo, P. Marin, D. Martin, L. Marcenaro, A. Cavallaro, and C. S. Regazzoni, "A multi-perspective approach to anomaly detection for self-aware embodied agents," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6598–6602.

[7] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 613–626, 2018.

[8] D. Fassbender, B. C. Heinrich, T. Luettel, and H.-J. Wuensche, "An optimization approach to trajectory generation for autonomous vehicle following," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3675–3680.

[9] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, and I. Matthews, "Win at home and draw away: Automatic formation analysis highlighting the differences in home and away team behaviors," in *Proceedings of 8th annual MIT sloan sports analytics conference*. Citeseer, 2014, pp. 1–7.

[10] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[11] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas *et al.*, "Reinforcement and imitation learning for diverse visuomotor skills," *arXiv preprint arXiv:1802.09564*, 2018.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[13] A. Edwards, C. Isbell, and A. Takanishi, "Perceptual reward functions," *arXiv preprint arXiv:1608.03824*, 2016.

[14] A. Y. Ng, S. J. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[15] T. Munzer, B. Piot, M. Geist, O. Pietquin, and M. Lopes, "Inverse reinforcement learning in relational domains," in *International Joint Conferences on Artificial Intelligence*, 2015.

[16] M. Baydoun, D. Campo, D. Kanapram, L. Marcenaro, and C. S. Regazzoni, "Prediction of multi-target dynamics using discrete descriptors: an interactive approach," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3342–3346.

[17] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang, "Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5999–6008, 2017.

[18] Y. Feng, J. Sun, and P. Chen, "Vehicle trajectory reconstruction using automatic vehicle identification and traffic count data," *Journal of advanced transportation*, vol. 49, no. 2, pp. 174–194, 2015.

[19] X. Sun, N. H. Yung, and E. Y. Lam, "Unsupervised tracking with the doubly stochastic dirichlet process mixture model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2594–2599, 2016.

[20] J. C. Nascimento, M. A. Figueiredo, and J. S. Marques, "Activity recognition using a mixture of vector fields," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1712–1725, 2012.

[21] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[22] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 425–437, 2002.

[23] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[24] L. E. Sucar, "Probabilistic graphical models," *Advances in Computer Vision and Pattern Recognition. London: Springer London. doi*, vol. 10, pp. 978–1, 2015.

[25] Z. Ghahramani, "Learning dynamic bayesian networks," in *International School on Neural Networks, Initiated by IIASS and EMFCSC*. Springer, 1997, pp. 168–197.

[26] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, "Learning switching models for abnormality detection for autonomous driving," in *2018 21st International Conference on Information Fusion (FUSION)*, July 2018, pp. 2606–2613.

[27] Y. Zheng, S. Jia, Z. Yu, T. Huang, J. K. Liu, and Y. Tian, "Probabilistic inference of binary markov random fields in spiking neural networks through mean-field approximation," *Neural Networks*, 2020.

[28] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, "Learning switching models for abnormality detection for autonomous driving," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 2606–2613.

[29] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.