Efficient Multi-Instance Learning for Activity Recognition from Time Series Data Using an Auto-Regressive Hidden Markov Model

Xinze Guan, Raviv Raich, Weng-Keen Wong {GUAN, RAICH, WONG}@EECS.OREGONSTATE.EDU
School of EECS, Oregon State University, Corvallis, OR 97331-5501 USA

Abstract

Activity recognition from sensor data has spurred a great deal of interest due to its impact on health care. Prior work on activity recognition from multivariate time series data has mainly applied supervised learning techniques which require a high degree of annotation effort to produce training data with the start and end times of each activity. In order to reduce the annotation effort, we present a weakly supervised approach based on multi-instance learning. We introduce a generative graphical model for multi-instance learning on time series data based on an auto-regressive hidden Markov model. Our model has a number of advantages, including the ability to produce both bag and instance-level predictions as well as an efficient exact inference algorithm based on dynamic programming.

1. Introduction

Modern wearable sensors, such as accelerometers, have provided an unobtrusive and cost-effective way to accurately measure body movement. In a "free-living" environment (i.e. outside a controlled laboratory environment), the individual wearing the sensor performs a diverse set of variable-length activities during their daily routine and the data from these activities is collected. Typically, this data is in the form of a continuous multivariate time series. This data can then be used to perform physical activity recognition (PAR), which involves segmenting the time series into individual activities and then identifying the individual activities. PAR is an important task in health surveillance and epidemiological research (Bauman et al., 2006) as well as in assistance of individuals with cognitive disorders (Popescu & Mahnot, 2012).

Prior approaches to PAR have mainly employed supervised learning techniques (see Section 2). These approaches require labeled sensor data for training, with the labels identifying the activity as well as the start and end times of each activity as shown in Figure 1 (top). A viable annotation approach under free-living conditions is *experience sampling* (Froehlich et al., 2007), which periodically prompts the user to annotate their activities since their previous prompt.

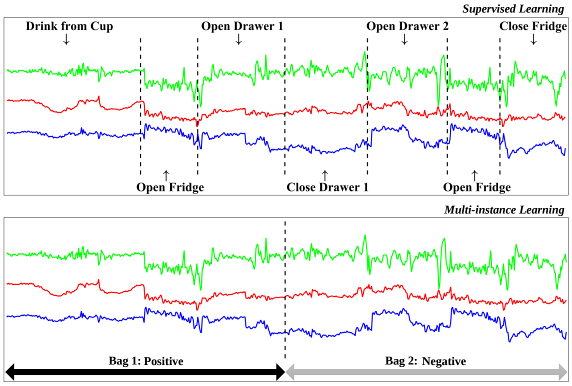

Figure 1. (Top) A synthesized sample of 3 dimensional accelerometer data created from the Opportunity data set (Chavarriaga et al., 2013). Each activity in this sequence is labeled, along with its start and end times. The data from each axis is shown in a different color. (Bottom) An example of a positive and a negative bag in multi-instance learning applied to PAR, with "Drink from Cup" as the activity of interest.

In order to reduce annotation effort in experience sampling, (Stikic et al., 2011) proposed a weakly supervised approach based on multi-instance learning (MIL) (Dietterich et al., 1997). In MIL, instead of labeling all instances, the annotator only needs to label bags of instances. A bag is labeled positive if and only if there is at least one positive instance in the bag. The ambiguity of the positive bags reduces the labeling effort but puts the burden of resolving the ambiguity on the learning algorithm. When MIL is applied to activity recognition data, the bags correspond to data in the time interval between experience sampling prompts. Although these intervals (bags) typically contain multiple activities, under a MIL setting, the bags are la-

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

beled as positive or negative. The meaning of the bag label can vary depending on the bag labeling scheme. For instance, a bag could be given a positive label if the majority of the time interval is spent doing a specific activity of interest (e.g. “Drink from Cup”), as shown in Figure 1 (bottom). We adopt this majority labeling scheme in our work. Other bag labeling schemes were proposed in (Stikic et al., 2011). Note that the length of the interval between experience sampling prompts plays an important role in the MIL setting. With longer intervals, the user needs to provide fewer labels. However, the bags will contain more data and the ambiguity in the labeling also increases.

We build on the work by (Stikic et al., 2011) and propose a novel MIL model for offline activity recognition from multivariate time series data. Our model is a generative graphical model based on an Auto-Regressive Hidden Markov Model (ARHMM), which improves on the approach by (Stikic et al., 2011) by modeling the temporal dynamics of the time series. Since the model is generative, we can predict both bag labels as well as instance labels. Finally, we remedy a naive approach to training our model that results in a run time that is exponential in the number of instances in a bag. We show how to use dynamic programming to reduce the time complexity to be quadratic in the number of instances in a bag. We evaluate our approach on real-world multivariate sensor data sets and show that it consistently performs well compared to other approaches.

2. Related Work

Prior work on PAR from time series data has mostly taken a supervised learning approach, which assumes the training data is fully labeled with the start and end times of each activity. A common strategy is the sliding window approach (Dietterich, 2002), which slides a window along the time series and converts the data in the window into a feature vector. Common feature representations that have been used include statistical features of the raw signal (e.g., mean and variance) and frequency domain features. With this new representation of the data, a standard supervised learning algorithm can be then applied (e.g., see (Bao & Intille, 2004; Ravi et al., 2005; Zheng et al., 2013)). Other techniques applied to PAR include sequential learning algorithms capable of modeling sequential relationships (e.g. (Lester et al., 2005; van Kasteren et al., 2008; Wu et al., 2009)) and approaches that discover subsequences of the original time series that are predictive of the class label (Ye & Keogh, 2009; Hu et al., 2013). All of these techniques have been developed for a standard supervised learning setting rather than a MIL setting and thus require a high degree of annotation effort by the user to produce fully labeled training data.

2.1. Multi-instance Learning

MIL (Dietterich et al., 1997) provides a weakly supervised alternative that can alleviate the annotation burden on the user. In MIL, the training data consists of B bags. The bth bag can be represented as a tuple $(\mathbf{X}_b, Y_b)$, where $\mathbf{X}_b = \{\mathbf{X}_b^1, \ldots, \mathbf{X}_b^{N_b}\}$ is a set of N_b instances and Y_b is the bag-level label (positive or negative). Note that an instance is represented in boldface because it is represented as a multivariate feature vector. Each instance has an associated instance-level label (positive or negative) that is considered hidden. Under the standard “presence-based” assumption (Weidmann et al., 2003), a bag is labeled positive if there is at least one instance that is positive and negative if none of the instances are positive. The main task of a MIL algorithm is to predict the bag-level label given the instance level features and without being given the instance-level labels.

The literature on MIL is extensive and we focus on the most directly related work in this section. Several approaches have proposed using graphical models (e.g., (Adel et al., 2013; Hajimirsadeghi et al., 2013)) for MIL. The most related work are by (Foulds & Smyth, 2011) and (Kandemir & Hamprecht, 2014). Both approaches use a mixture model to discover mixture components that are predictive of the bag label. However, unlike in our work, the instances are assumed to be i.i.d. feature vectors and they do not model relationships between mixture components.

To our knowledge, the first work that has applied MIL to activity recognition from time series data was by (Stikic et al., 2011). In their work, they applied a sliding window to a time series to create instances, which were produced by converting the data within the window to a fixed-length feature vector consisting of statistical features of the raw signal and FFT coefficients. The mi-SVM algorithm (Andrews et al., 2003) was trained on these bags. Recently, MIL algorithms have been developed for *structured* data, capturing relationships between instances (Zhou et al., 2009; Warrell & Torr, 2011), relationships between bags (Zhang et al., 2011), or relationships between instances in different bags (Deselaers & Ferrari, 2010). Our approach is the first to model the temporal dynamics between instances in a bag.

2.2. Autoregressive Models

Autoregressive (AR) models (Hamilton, 1994) capture the temporal structure of time series data by assuming that the current observation x^t is a weighted linear combination of the previous p observations $x^{t-p}, \ldots, x^{t-2}, x^{t-1}$ (or $x^{(t-p):(t-1)}$ in shorthand notation). The Vector AR (VAR) model is a generalization of the univariate AR model to multivariate observations. The d-dimensional observation of a VAR model of order p, which we denote as $\mathbf{X}^t$ follows

$$\mathbf{X}^t = \mathbf{A}_0 + \sum_{i=1}^{p} \mathbf{A}_i \mathbf{X}^{t-i} + \boldsymbol{\epsilon}^t \tag{1}$$

where $\mathbf{A}_0$ is a d-dimensional intercept vector, $\mathbf{A}_i$'s are $d \times d$ matrices corresponding to the weights in the linear combination of past observations, and $\boldsymbol{\epsilon}^t$ is a d-dimensional white noise term drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Alternatively, we can write $\mathbf{X}^t | \mathbf{X}^{(t-p):(t-1)} \sim \mathcal{N}(\mathbf{A_0} + \sum_{i=1}^{p} \mathbf{A}_i \mathbf{X}^{t-i}, \boldsymbol{\Sigma})$. The dependence on p past values of the process allows for a convenient graphical model representation.

2.3. Mixture of AR processes

A K-component mixture of VAR models (MAR) can be created from K separate VAR models by randomly selecting the output of one component following a discrete distribution with parameters $(\psi_1, \ldots, \psi_K)$ which represent the mixture weights (Wong & Li, 2000). We compute the conditional probability of $\mathbf{X}^t$ given the previous p readings as:

$$P(\mathbf{X}^t | \mathbf{X}^{(t-p):(t-1)}) = \sum_{k=1}^{K} \psi_k \mathcal{N}(\mathbf{A}_{k0} + \sum_{i=1}^{p} \mathbf{A}_{ki} \mathbf{X}^{t-i}, \boldsymbol{\Sigma}_k)$$

where $\psi_1 + \ldots + \psi_K = 1$ and $\psi_k > 0$, for $k = 1, \ldots, K$. MAR can be fit to data using Expectation-Maximization (EM) (Dempster et al., 1977).

3. Methodology

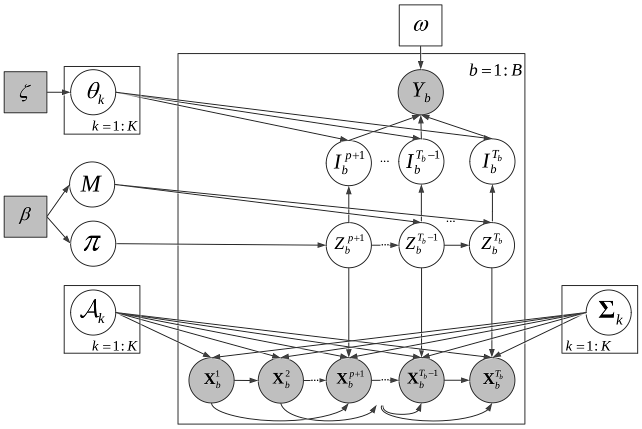

Figure 2. The graphical model representation of the ARHMM-MIL model

Figure 2 shows the Auto-Regressive Hidden Markov Model for Multiple Instance Learning (ARHMM-MIL) as a graphical model. For illustrative simplicity, we depict the ARHMM-MIL Model with an AR process of order 2. We assume that bags (i.e. time series) are independent and that each time-series is obtained by the following generative model.

Mixture Components. For time series (bag) b, at each time-instance $t = p + 1, \ldots, T_b$, a mixture component index $Z_b^t \in \{1, 2, \cdots, K\}$ is generated according to a chain model described by the following two steps. First an initial mixture component index Z_b^{p+1} is generated following the discrete probability model

$$P(Z_b^{p+1} = j | \boldsymbol{\pi}) = \pi_j, \tag{2}$$

where $\pi_j \geq 0$ and $\sum_j \pi_j = 1$. Then, the remaining values of Z_b^t for $t = p + 2, \ldots, T_b$ are generated in a recursive fashion according to the following transition probability:

$$P(Z_b^t = j | Z_b^{t-1} = i, \mathbf{M}) = M_{ij}, \tag{3}$$

where $\mathbf{M}$ is a transition probability matrix satisfying $M_{ij} \geq 0$ and $\sum_j M_{ij} = 1$. The ARHMM-MIL model takes the transition between the hidden states Z_b^t into account, encouraging the consistency in adjacent hidden states. We consider both $\mathbf{M}$ and $\boldsymbol{\pi}$ to be random vectors. The vector $\boldsymbol{\pi}$ and the rows of $\mathbf{M}$ are probability vectors that are generated from a Dirichlet distribution with hyperparameter vector $\boldsymbol{\beta}$. Rather than learn the hyperparameter $\boldsymbol{\beta}$ from data, we assign it a relatively small value so that the resulting Dirichlet distribution will place the majority of the probability mass around a few mixture components.

Auto-regressive observation model. Given the mixture component $Z_b^t = k$, the corresponding time-series observation vector $\mathbf{X}_b^t$ is generated following a VAR(p) model parametrized by the kth component AR coefficients, i.e.,

$$\mathbf{X}_b^t | \mathbf{X}_b^{(t-p):(t-1)}, Z_b^t = k \sim \mathcal{N}(\mathbf{A}_{k0} + \sum_{i=1}^{p} \mathbf{A}_{ki} \mathbf{X}_b^{t-i}, \boldsymbol{\Sigma}_k), \tag{4}$$

for $t = p + 1, \ldots, T_b$, where $\mathbf{A}_{k0}$ is a d-dimensional intercept vector for the kth mixture component, $\mathbf{A}_{ki}$ a $d \times d$ weighting matrix corresponding to the ith past vector for the kth mixture component, and $\boldsymbol{\Sigma}_k$ is the $d \times d$ covariance matrix for the kth mixture component. Here, we assume that the first p observations $\mathbf{X}_b^1, \ldots, \mathbf{X}_b^p$ follow a joint distribution, $P(\mathbf{X}_b^1, \ldots, \mathbf{X}_b^p)$ that is independent of any of parameters of the proposed model. We denote the set of linear coefficient matrices for the kth component as $\boldsymbol{\mathcal{A}}_k = \{\mathbf{A}_{0k}, \ldots, \mathbf{A}_{pk}\}$. Likewise, we denote $\boldsymbol{\mathcal{A}} = \{\boldsymbol{\mathcal{A}}_1, \ldots, \boldsymbol{\mathcal{A}}_K\}$ and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K\}$. Thus, the Z_b^t and $\mathbf{X}_b^t$ layers form an Auto-Regressive HMM (Juang & Rabiner, 1985).

Label generation Given the mixture component index Z_b^t, an instance-level label $I_b^t \in \{0, 1\}$ is Bernoulli random variable $I_b^t | Z_b^t = k \sim$ Bernoulli(θ_k), i.e.,

$$P(I_b^t = l | Z_b^t = k, \boldsymbol{\theta}) = \theta_k^l (1 - \theta_k)^{1-l}, \tag{5}$$

where θ_k denotes the probability that the label of the kth component is positive. The parameter θ_k is a random variable which has a Beta prior parameterized by $\boldsymbol{\zeta}$. As with $\boldsymbol{\beta}$, we assign hyperparameter $\boldsymbol{\zeta}$ a pre-determined value that sets all of its entries to a low value to encourage a high probability for either label value 0 or 1.

Our proposed model relaxes the union rule (Foulds & Frank, 2010) or the *presence-based* assumption (Weidmann et al., 2003) using the following probabilistic relation between the bag label and the instance labels. Based on a variant of the soft-max function used in (Maron, 1998), given all instance level labels $I_b^{p+1}, \ldots, I_b^{T_b}$, the observed bag-level label Y_b is generated by:

$$P(Y_b = 1|\mathbf{I}_b) = \frac{\sum_{t=p+1}^{T_b} I_b^t \exp(\omega I_b^t)}{\sum_{t=p+1}^{T_b} \exp(\omega I_b^t)}, \tag{6}$$

where ω is a model parameter. When $\omega \to \infty$, the bag label is positive if at least one instance in the bag is positive and the bag is negative if all of the instances are negative. When $\omega \to 0$, the probability that the bag label is positive increases as the proportion of positive instances in the bag increases.

3.1. Parameter Estimation

We now show how to learn the model parameters $\boldsymbol{\Theta} = \{\boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\pi}, \mathbf{M}, \boldsymbol{\mathcal{A}}, \boldsymbol{\Sigma}\}$ from a collection of training bags. Since a prior is provided for some parameters, we solve the problem using the maximum-a-posterior (MAP) framework. Under this framework, the goal is to maximize the sum of the log-likelihood and the log of the prior, i.e., $\max_{\boldsymbol{\Theta}} \ell_{inc}(\boldsymbol{\Theta}) + \log P(\boldsymbol{\Theta})$. Since a prior is only provided for $\boldsymbol{\pi}$, $\mathbf{M}$ and $\boldsymbol{\theta}$, we consider a uniform prior over the rest of the parameters and hence $P(\boldsymbol{\Theta}) = P(\boldsymbol{\pi}|\boldsymbol{\beta})P(\mathbf{M}|\boldsymbol{\beta})P(\boldsymbol{\theta}|\zeta)$. Similar to maximum likelihood (ML), MAP can be solved using Expectation-Maximization by replacing the incomplete log-likelihood $\ell_{inc}(\boldsymbol{\Theta})$ with the auxiliary function $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}')$ yielding the following iteration: $\boldsymbol{\Theta}^{k+1} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^k) + \log P(\boldsymbol{\Theta})$. This approach can be extended to the online setting by using the approach of (Cappé, 2011).

The auxiliary function is derived using the complete-data log-likelihood. The observed data in our model is given by $\{\mathbf{X}_b, Y_b\}$ for $b = 1, \ldots, B$ while the complete data is given by $\{\mathbf{Z}_b, \mathbf{I}_b, \mathbf{X}_b, Y_b\}$ for $b = 1, \ldots, B$. The complete-data log-likelihood is as follows:

$$\begin{aligned}\ell(\boldsymbol{\Theta}) = \sum_{b=1}^{B} \Bigg[& \log P(Y_b|\mathbf{I}_b, \omega) + \sum_{t=p+1}^{T_b} \log P(I_b^t|Z_b^t, \boldsymbol{\theta}) \\ & + \log P(Z_b^{p+1}|\boldsymbol{\pi}) + \sum_{t=p+2}^{T_b} \log P(Z_b^t|Z_b^{t-1}, \mathbf{M}) \\ & + \sum_{t=p+1}^{T_b} \log P(\mathbf{X}_b^t|\mathbf{X}_b^{(t-p):(t-1)}, Z_b^t, \boldsymbol{\mathcal{A}}, \boldsymbol{\Sigma}) \Bigg], \end{aligned} \tag{7}$$

where $P(Y_b|\mathbf{I}_b, \omega)$, $P(I_b^t|Z_b^t, \boldsymbol{\theta})$, $P(Z_b^t|Z_b^{t-1}, \mathbf{M})$, $P(Z_b^{p+1}|\boldsymbol{\pi})$, and $P(\boldsymbol{X}_b^t|\boldsymbol{X}_b^{(t-p):(t-1)}, Z_b^t, \boldsymbol{\mathcal{A}}, \boldsymbol{\Sigma})$ are given by (2)-(6). Based on the complete-data log-likelihood in (7), we obtain the auxiliary function $\mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}')$ by taking the expected value of the complete data log-likelihood $\ell(\boldsymbol{\Theta})$ with respect to the latent variables $\mathbf{Z}$, $\mathbf{I}$ conditioned on observations $\mathbf{X}$, bag labels Y and the model parameters $\boldsymbol{\Theta}'$, i.e., using $P(\mathbf{Z}, \mathbf{I}|\mathbf{X}, Y, \boldsymbol{\Theta}')$:

$$\begin{aligned} & \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta'}) \\ & = \sum_{b=1}^{B} \sum_{\mathbf{I}} P(\mathbf{I}_b = \mathbf{I}|\mathbf{X}_b, Y_b, \boldsymbol{\Theta}') \Big(Y_b \log P(Y_b = 1|\mathbf{I}_b, \omega) \\ & \quad + (1 - Y_b) \log P(Y_b = 0|\mathbf{I}_b, \omega) \Big) \\ & \quad + \sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \sum_{l=0}^{1} \sum_{j=1}^{K} P(I_b^t = l, Z_b^t = j|\mathbf{X}_b, Y_b, \boldsymbol{\Theta}') \\ & \quad \cdot \log P(\mathbf{I}_b^t = l|Z_b^t = j, \boldsymbol{\theta}) \\ & \quad + \sum_{b=1}^{B} \sum_{j=1}^{K} P(Z_b^{p+1} = j|\mathbf{X}_b, Y_b, \boldsymbol{\Theta'}) \log P(Z_b^{p+1} = j|\boldsymbol{\pi}) \\ & \quad + \sum_{b=1}^{B} \sum_{t=p+2}^{T_b} \sum_{j=1}^{K} \sum_{i=1}^{K} P(Z_b^t = j, Z_b^{t-1} = i|\mathbf{X}_b, Y_b, \boldsymbol{\Theta'}) \\ & \quad \cdot \log P(Z_b^t = j|Z_b^{t-1} = i, \mathbf{M}) \\ & \quad + \sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \sum_{j=1}^{K} P(Z_b^t = j|\mathbf{X}_b, Y_b, \boldsymbol{\Theta}') \\ & \quad \cdot \log P(\mathbf{X}_b^t|\mathbf{X}_b^{(t-p):(t-1)}, Z_b^t = j, \boldsymbol{\mathcal{A}}, \boldsymbol{\Sigma}). \end{aligned} \tag{8}$$

Details of the derivation are in the supplementary material. The auxiliary function $\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}')$ decomposes into five terms, with each term depending on a different set of parameters allowing for a convenient estimation of the individual parameters.

3.2. M-Step

The maximization of (8) in addition to $\log P(\boldsymbol{\Theta})$ w.r.t. to each of of the terms in $\boldsymbol{\Theta} = \{\omega, \boldsymbol{\theta}, \boldsymbol{\pi}, \mathbf{M}, \boldsymbol{\mathcal{A}}, \boldsymbol{\Sigma}\}$ yields the update rule for each of the parameters as shown below (with details in the supplementary material).

1. The update rule for π_j is:

$$\pi_j = \frac{\sum_{b=1}^{B} P(Z_b^{p+1} = j|\mathbf{X}_b, Y_b, \boldsymbol{\Theta}') + \beta_j - 1}{\sum_{j'=1}^{K} \sum_{b=1}^{B} P(Z_b^{p+1} = j'|\mathbf{X}_b, Y_b, \boldsymbol{\Theta}') + \beta_{j'} - 1} \tag{9}$$

2. Let $\rho_{bt}^{ij} = P(Z_b^t = j, Z_b^{t-1} = i | \mathbf{X}_b, Y_b, \mathbf{\Theta}')$. The update rule for the transition matrix M_{ij} is:

$$M_{ij} = \frac{\sum_{b=1}^{B} \sum_{t=p+2}^{T_b} \rho_{bt}^{ij} + \beta_j - 1}{\sum_{j'=1}^{K} \sum_{b=1}^{B} \sum_{t=p+2}^{T_b} \rho_{bt}^{ij'} + \beta_{j'} - 1} \tag{10}$$

3. The generalization of Yule-Walker equations can be used to estimate the AR parameters. Let $\xi_{btj} = P(Z_b^t = j | \mathbf{X}_b, Y_b, \mathbf{\Theta}')$. The update rule for $\boldsymbol{\mathcal{A}}$ is given by the solution to the following set of linear equations:

$$\sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \xi_{btj} \left[\mathbf{A}_{j0} + \sum_{s=1}^{p} \mathbf{A}_{js} (\mathbf{X}_b^{t-s}) \right] = \sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \xi_{btj} (\mathbf{X}_b^t) \tag{11}$$

$$\sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \xi_{btj} \left[\mathbf{A}_{j0} (\mathbf{X}_b^{t-k})' + \sum_{s=1}^{p} \mathbf{A}_{js} (\mathbf{X}_b^{t-s}) (\mathbf{X}_b^{t-k})' \right] = \sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \xi_{btj} (\mathbf{X}_b^t) (\mathbf{X}_b^{t-k})' \quad \text{for } k = 1, \ldots, p \tag{12}$$

4. Let $\boldsymbol{\mu}_{btj} = \mathbf{X}_b^t - (\mathbf{A}_{j0} + \sum_{k=1}^{p} \mathbf{A}_{jk} \mathbf{X}_b^{t-k})$. The update rule for the covariance matrix of jth mixture component $\mathbf{\Sigma}_j$ is:

$$\mathbf{\Sigma}_j = \frac{\sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \xi_{btj} \boldsymbol{\mu}_{btj} (\boldsymbol{\mu}_{btj})'}{\sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \xi_{btj}} \tag{13}$$

5. Let $\delta_{btj}^{l} = P(I_b^t = l, Z_b^t = j | \mathbf{X}_b, Y_b, \mathbf{\Theta}')$. The update rule for θ_j is given by:

$$\theta_j = \phi_j / (1 + \phi_j) \tag{14}$$

where

$$\phi_j = \frac{\sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \delta_{btj}^{1} + \zeta_1 - 1}{\sum_{b=1}^{B} \sum_{t=p+1}^{T_b} \delta_{btj}^{0} + \zeta_2 - 1}.$$

6. Since there is no closed-form solution for updating ω, a Newton iteration is used to update ω through the equation $\omega = \omega' - l'(\omega')/l''(\omega')$, where $l'(\omega)$ and $l''(\omega)$ are the first and second derivatives of $\mathcal{Q}(\mathbf{\Theta}; \mathbf{\Theta}')$ w.r.t. to ω.

To simplify the derivative computations, we define $N_b^t = \sum_{t'=p+1}^{t} I_b^{t'}$, which is the number of positive instances in bag b up to the tth instance. The positive bag probability $P(Y_b = 1 | \mathbf{I}_b)$ can be equivalently computed using:

$$P(Y_b = 1 | N_b^{T_b}) = \frac{N_b^{T_b} \exp(\omega)}{N_b^{T_b} \exp(\omega) + T_b - p - N_b^{T_b}} \tag{15}$$

We compute the first and second derivative as follows.

$$l'(\omega) = \sum_{b=1}^{B} \left[Y_b - \sum_{C=0}^{T_b - p} P(N_b^{T_b} = C | \mathbf{X}_b^{1:T_b}, Y_b) \cdot P(Y_b = 1 | N_b^{T_b} = C) \right] \tag{16}$$

$$l''(\omega) = -\sum_{b=1}^{B} \sum_{C=0}^{T_b - p} P(N_b^{T_b} = C | \mathbf{X}_b^{1:T_b}, Y_b) \cdot P(Y_b = 1 | N_b^{T_b} = C) P(Y_b = 0 | N_b^{T_b} = C) \tag{17}$$

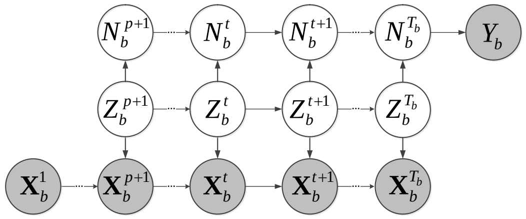

Figure 3. A graphical model representing a bag in Figure 2 with the instance label I replaced by a counting variable N.

3.3. E-step via Efficient Message Passing

The auxiliary function (8) requires the following three terms:

1. $P(\sum_{t=p+1}^{T_b} I_b^t = C | \mathbf{X}_b, Y_b, \mathbf{\Theta}')$, this term is used to update ω (see (16)-(17));

2. $P(I_b^t = l, Z_b^t = j | \mathbf{X}_b, Y_b, \mathbf{\Theta}')$, this term is used for the update of $\boldsymbol{\theta}$ (see (14)); and

3. $P(Z_b^t = j, Z_b^{t-1} = i | \mathbf{X}_b, Y_b, \mathbf{\Theta}')$, this term is used to update $\mathbf{M}$ (see (10)) as well as to derive the marginal $P(Z_b^t = j | \mathbf{X}_b, Y_b, \mathbf{\Theta}') = \sum_{i=1}^{K} P(Z_b^t = j, Z_b^{t-1} = i | \mathbf{X}_b, Y_b, \mathbf{\Theta}')$, which is used in the updates of $\boldsymbol{\pi}$ (see (9)) and the AR model parameters $\boldsymbol{\mathcal{A}}$ (see (11), (12)) and $\mathbf{\Sigma}$ (see (13)).

These three probabilities are non-trivial to calculate due to dependence between random variables created by conditioning on the label Y_b. To address this challenge, we first propose a reformulation of the graphical model as a chain by replacing the sequence I_b^t with its cumulative sum N_b^t

(as done in (16) and (17)). The N_b^t's can be used to compute probabilities associated with the nodes I_b^t using the following relation $I_b^t = N_b^t - N_b^{t-1}$. Using the N_b^t's and the Z_b^t's the three probability terms can be replaced with

1. $P(N_b^t = C|\mathbf{X}_b, Y_b, \mathbf{\Theta}')$, where $t = T_b$
2. $P(N_b^t - N_b^{t-1} = l, Z_b^t = j|\mathbf{X}_b, Y_b, \mathbf{\Theta}')$, and
3. $P(Z_b^t = j, Z_b^{t-1} = i|\mathbf{X}_b, Y_b, \mathbf{\Theta}')$.

We aim to compute $P(N_b^t, Z_b^t, N_b^{t-1}, Z_b^{t-1}|\mathbf{X}_b, Y_b, \mathbf{\Theta}')$ from which the above three probability terms can be computed. While not obvious, the reformulated graphical model in Figure 3 offers a chain that lends itself to an efficient forward-backward message passing approach. To make this more obvious, we replace (N_b^t, Z_b^t) with a supernode $S_t^b = (N_b^t, Z_b^t)$, as seen in Figure 4.

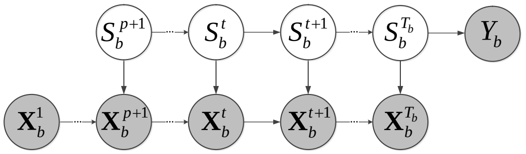

Figure 4. A simplified graphical model by representing the (N, Z) variables pair as a supernode S.

3.3.1. Message passing for a generalized version of a chain model

To aid with exposition, we first introduce the message-passing algorithm on the simplified graphical model in Figure 4 and we omit the conditioning on the parameter set $\mathbf{\Theta}'$. The message passing algorithm first sends a forward message which is computed while traversing the graphical model from left to right. Then, a backward message is computed while traversing the graphical model from right to left. Finally the messages are used to form the pairwise probability for (S_b^t, S_b^{t-1}) conditioned on the observed nodes $\mathbf{X}_b^1, \ldots, \mathbf{X}_b^{T_b}$ and Y_b.

Specifically, the **forward message** is defined as $\alpha_b^q(t) = P(\mathbf{X}_b^{1:t}, S_b^t = q)$. It is initialized by computing $\alpha_b^q(p{+}1) = P(\mathbf{X}_b^{1:p+1}, S_b^t = q)$ and is computed recursively using

$$\alpha_b^q(t) = \sum_p \alpha_b^p(t-1) P(S_b^t = q|S_b^{t-1} = p)$$
$$\cdot P(\mathbf{X}_b^t|\mathbf{X}_b^{1:t-1}, S_b^t = q) \qquad \text{for } t = p+2, \ldots, T_b.$$

The **backward message** is defined by $\beta_b^q(t) = P(Y_b, \mathbf{X}_b^{t+1:T_b}|\mathbf{X}_b^{1:t}, S_b^t = q)$. It is initialized by setting $\beta_b^q(T_b) = P(Y_b|\mathbf{X}_b^{1:T_b}, S_b^{T_b} = q)$ and is computed recursively by

$$\beta_b^q(t) = \sum_r P(S_b^{t+1} = r|S_b^t = q) P(\mathbf{X}_b^{t+1}|\mathbf{X}_b^{1:t}, S_b^{t+1} = r)$$
$$\cdot \beta_b^r(t+1) \quad \text{for } t = T_b - 1, \ldots, p+1.$$

As the beginning of the section suggests, the E-step necessitates the computation of $P(S_b^t = q, S_b^{t-1} = r|\mathbf{X}_b, Y_b)$. To compute this conditional probability, we focus on computing the joint distribution $P(S_t = q, S_{t-1} = r, \mathbf{X}_b, Y_b)$. This joint distribution can be written in term of the forward message, the backward message, the state transition probability, and the observation model probability as

$$P(S_b^t = q, S_b^{t-1} = r, \mathbf{X}_b, Y_b) = P(S_b^t = q|S_b^{t-1} = r)$$
$$\cdot \beta_b^q(t) P(\mathbf{X}_b^t|\mathbf{X}_b^{t-p:t-1}, S_b^t = q) \alpha_b^r(t-1) \quad (18)$$

Although the forward and backward messages resemble those of an ARHMM, the sparsity of the transition matrix between S_{t-1} and S_t makes inference in our ARHMM-MIL model a special case of message passing. With this special case, exact inference is efficient, which we will show in the following section.

3.3.2. Message passing for the chain based on (N_b^t, Z_b^t)

We now expand the S_b^t nodes into the components N_b^t, and Z_b^t. Full derivations for the equations below are in the supplementary material. Let $q = (q_N, q_Z)$ and $r = (r_N, r_Z)$. Since we assume that the first p observations $\boldsymbol{X}_b^1, \ldots, \boldsymbol{X}_b^p$ follow a joint distribution that is independent of any of the proposed model parameters, the forward message is initialized for $q_N \in \{0, 1\}$ with

$$\alpha_b^{q_N,q_Z}(p+1) = P(\mathbf{X}_b^{1:p+1}, N_b^{p+1} = q_N, Z_b^{p+1} = q_Z)$$
$$= A_b^{q_Z}(p+1)(\theta_{q_Z})^{q_N}(1-\theta_{q_Z})^{(1-q_N)}\pi_{q_Z} P(\mathbf{X}_b^{1:p})$$

and zero otherwise. Then, the forward message is recursively computed for $t = p+2, \ldots, T_b$ using

$$\alpha_b^{q_N,q_Z}(t) = A_b^{q_Z}(t) \sum_{r_Z} \sum_{r_N} \bigg(\mathbb{I}(q_N = r_N)(1-\theta_{q_Z})$$
$$+ \mathbb{I}(q_N = r_N + 1)\theta_{q_Z} \bigg) M_{r_Z,q_Z} \alpha_b^{r_N,r_Z}(t-1)$$

where $\pi_{q_Z} = P(Z_b^{p+1} = q_Z)$, $M_{r_Z,q_Z} = P(Z_b^t = q_Z|Z_b^{t-1} = r_Z)$, $A_b^{q_Z}(t) = P(\mathbf{X}_b^t|\mathbf{X}_b^{1:t-1}, Z_b^t = q_Z)$ and $\theta_{q_Z} = P(I_b^t = 1|Z_b^t = q_Z)$. The summation over r_N involves only two non-zero terms ($q_N = r_N$ and $q_N = r_N + 1$), while the summation over r_Z involves K terms. Thus, the complexity of computing the recursive step is $O(K)$. For a bag b, the overall complexity for computing all messages $\alpha_b^{q_N,q_Z}(t)$ for $q_N = 0, \ldots, t-p$, $q_Z = 1, \ldots, K$ and $t = p+1, \ldots, T_b$ is $O(K^2 T_b^2)$.

The backward message is initialized with

$$\beta_b^{q_N,q_Z}(T_b) = P(Y_b|N_b^{T_b} = q_N)$$

for $q_Z = 1, \ldots, K$ and $q_N = 0, \ldots, T_b - p$, where $P(Y_b | N_b^{T_b} = q_N)$ is computed using (6). Then, the backward message is recursively computed for $t = T_b - 1, \ldots, p + 1$ using

$$\beta_b^{q_N, q_Z}(t) = \sum_{r_Z} M_{q_Z, r_Z} A_b^{r_Z}(t+1) \sum_{r_N} \Big(\mathbb{I}(r_N = q_N) \cdot (1 - \theta_{r_Z}) + \mathbb{I}(r_N = q_N + 1) \theta_{r_Z} \Big) \beta_b^{r_N, r_Z}(t+1)$$

where $M_{q_Z, r_Z} = P(Z_b^{t+1} = r_Z | Z_b^t = q_Z)$, $A_b^{r_Z}(t+1) = P(\mathbf{X}_b^{t+1} | \mathbf{X}_b^{1:t}, Z_b^{t+1} = r_Z)$ and $\theta_{r_Z} = P(I_b^{t+1} = 1 | Z_b^{t+1} = r_Z)$. The summation over r_N involves only two non-zero terms, which happens if $r_N = q_N$ or $r_N = q_N + 1$, while the summation over r_Z includes K terms. Hence, similar to the forward pass, the complexity of the recursive step is $O(K)$ and the total complexity for all the backward messages for a single bag is $O(K^2 T_b^2)$.

Finally, we can use the forward/backward probabilities to compute the pairwise state probability of (18):

$$\begin{aligned} &P(N_b^t = q_N, Z_b^t = q_Z, N_b^{t-1} = r_N, Z_b^{t-1} = r_Z, \mathbf{X}_b, Y_b) \\ &= \beta_b^{q_N, q_Z}(t) \theta_{q_Z}^{\mathbb{I}(q_N = r_N + 1)} (1 - \theta_{q_Z})^{\mathbb{I}(q_N = r_N)} M_{q_Z, r_Z} \\ &\quad \cdot A_b^{q_Z}(t) \alpha_b^{r_N, r_Z}(t-1). \end{aligned} \tag{19}$$

In summary, the total time complexity for computing (19) for a single bag b is $O(K^2 T_b^2)$, which is much more efficient than a naive approach which would take $O(2^{T_b} K^{T_b})$.

4. Experiment Results

The first dataset used in our evaluation is the *Opportunity* dataset (Chavarriaga et al., 2013), which is from an activity recognition task where we used the sensor data from a right wrist-worn accelerometer from three different subjects. We chose the middle-level activities by concatenating the daily living and drill runs together. The other two datasets are from the *Trainspotting* datasets (Berlin & Laerhoven, 2012), which involve a task analogous to PAR. These datasets contain raw acceleration data from a sensor node that was deployed on two different rail tracks, with the task of classifying the train (i.e. the activity) on the tracks.

For each dataset, we select a particular activity as the "positive" activity. Following our labeling scheme, a bag is labeled as positive if the majority of the observations come from this positive class and negative otherwise. We selected three activities from the *Opportunity* dataset (Open Fridge, Open Door 2, Close Drawer 3), corresponding to a frequent activity, a moderately frequent activity and an infrequent activity. For each of the *Trainspotting* datasets, we treat each type of train as the "positive activity", and we report results for all the three classes of trains for *Trainspotting 1*. We exclude 3-Wagon from *Trainspotting 2* because it is very infrequent, and report two other classes of trains. For all of our experiments, we used a bag size of 200 observations. Due to the imbalance in the number of positive and negative bags, we report bag and instance-level area under the ROC curve (AUC) instead of accuracy.

We compared ARHMM-MIL against three MIL algorithms: miSVM (Andrews et al., 2003) (with an RBF kernel), DPMIL (Kandemir & Hamprecht, 2014) and miGraph (Zhou et al., 2009), which models the relationship between instances in a bag. For these algorithms, we followed the experimental setup from (Stikic et al., 2011) and converted time series data into fixed-length feature vectors using the sliding window approach. This conversion was necessary because unlike ARHMM-MIL, these three algorithms do not model the time series data directly but require fixed-length feature vectors for the instances. We used the feature representation of (Stikic et al., 2011), which included both statistical and FFT features. We tuned the sliding window size over the value of $[3, 5, 7, 50, 100]$ to produce the best results; note that we included larger window sizes in our tuning range to provide more reliable estimates of the statistical and FFT features. Although miGraph (Zhou et al., 2009) could be used for instance-level annotation by treating the entire bag as one instance, we did not include these results as they were very poor.

In addition, we include results on a variant of the ARHMM-MIL model that does not model the sequential relationships between the AR processes. This variant removes the arrows between the Z_b^t nodes, resulting in a generative model that is a mixture of auto-regressive processes. We call this model the Mixture of Auto-Regressive processes Multi-Instance Learning (MARMIL) model. The MARMIL model also uses dynamic programming for efficient inference, but the inference process is simpler than that of ARHMM-MIL, which has the additional complexity of the Hidden Markov Model layer. Further details regarding the MARMIL model can be found in (Guan et al., 2015).

For our experimental results, we used a nested stratified cross validation. Stratification was necessary to ensure each fold had enough positive bags as there were more negative than positive bags. The outer 10-fold cross validation split the data into training and testing. The inner cross validation was used to tune parameters by splitting the training set into 5 folds. The parameters tuned included the order of the AR processes ($[3, 5, 7]$) and the number of components ($[5, 10, 15]$ for *Trainspotting* and $[10, 20, 30]$ for *Opportunity*) in the ARHMM-MIL and MARMIL models. For SVM-based methods, we tuned the C parameter ($[0.01, 0.1, 1]$)[1] and also the RBF kernel parameter

[1] We experimented with increasing the range for the C parameter but it had little effect on the results.

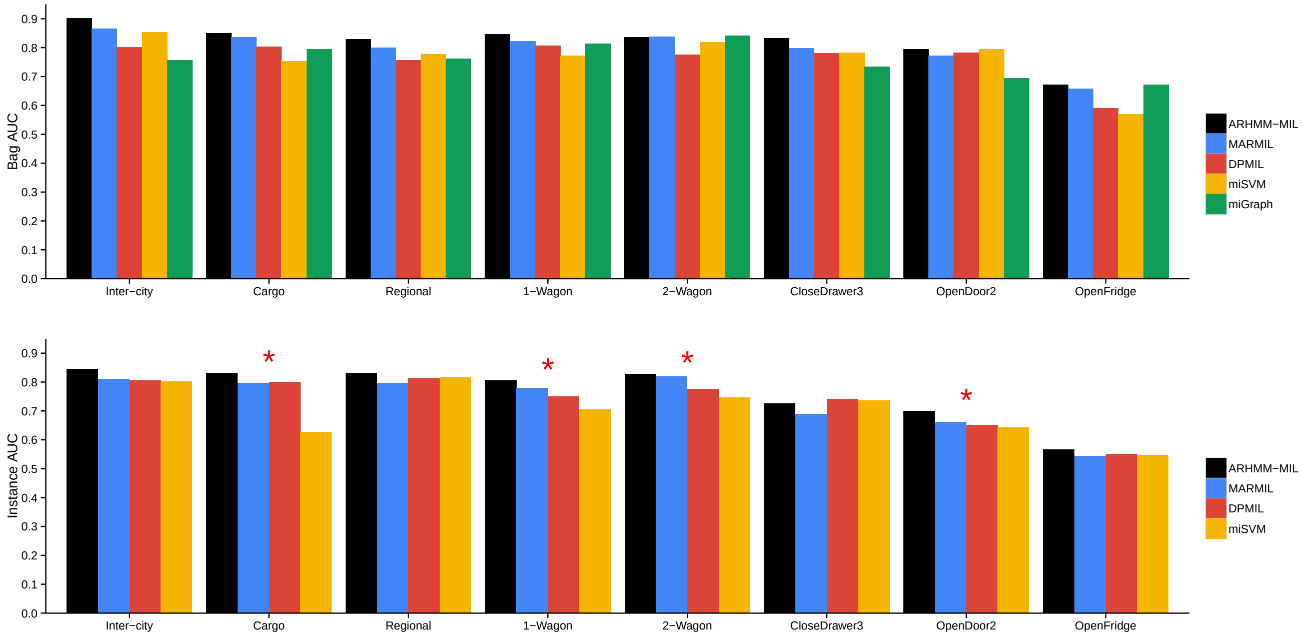

Figure 5. Average bag-level (top) and instance-level AUCs (bottom) for the Trainspotting1 (Inter-city, Cargo, Regional), Trainspotting2 (1-Wagon, 2-Wagon) and Opportunity (CloseDrawer3, OpenDoor2, OpenFridge) datasets. Statistically significant differences (paired Wilcoxon signed rank test, $\alpha = 0.05$) between ARHMM-MIL and all other algorithms are indicated by a star.

($[0.1, 1, 10]$). To reduce between-subject variability in the *Opportunity* dataset, we performed subject-specific experiments over subjects 1, 2 and 3 and averaged the results.

5. Discussion

Figure 5 shows the average bag-level AUC (top) and the instance-level AUC (bottom) for the various algorithms over all the datasets. The ARHMM-MIL model was consistently among the top performing algorithms; it had the best AUC in 6/8 datasets in terms of bag-level AUC and 7/8 datasets in terms of instance-level AUC, but the difference in AUC between ARHMM-MIL and all other algorithms is statistically significant in only a few datasets. In many cases, both ARHMM-MIL and MARMIL showed improvements over other methods because they can directly model the time series data as an auto-regressive process. The other algorithms are not able to do so as they operate on fixed-length feature vectors which capture coarse traits (i.e. statistics or FFT coefficients) of a window of data. The fact that ARHMM-MIL outperformed MARMIL in almost all cases demonstrates the benefit of modeling the transition between the AR processes. Although the miGraph algorithm can model the relationship between instances in a bag, for most of the datasets in our experiments, its graph kernel does not capture the temporal dynamics between instances as well as ARHMM-MIL.

We also highlight other important benefits of the ARHMM-MIL model. First, the instance-level AUC is fairly high in many of our experiments, indicating that even with ambiguously labeled data, we can identify parts of the time series that correspond to the activity of interest. Second, the ARHMM-MIL model allows us to decompose a time series into K auto-regressive processes and we can see how these processes transition to each other as activities occur. This capability allows a deeper analysis of the data beyond a simple class label prediction. Finally, the dynamic programming approach makes exact inference tractable and we refer the reader to the supplementary material for an empirical comparison of running time.

6. Conclusions

The ARHMM-MIL model is a generative graphical model that can predict both bag and instance labels through a tractable exact inference algorithm. When compared against state-of-the-art methods, the ARHMM-MIL model was consistently among the best performers due to its ability to model the relationship between instances. Our model also enables a deeper analysis of time series data by decomposing it into its component auto-regressive processes that are predictive of the bag label. For future work, we will extend our model to the multi-instance multi-label learning setting (Zhou et al., 2012).

Acknowledgements This work is partially supported by the National Science Foundation grants CCF-1254218.

References

Adel, T., Urner, R., Smith, B., Stashuk, D., and Lizotte, D. J. Generative multiple-instance learning models for quantitative electromyography. In *Proceedings on the 29th Conference on Uncertainty in Artificial Intelligence*, pp. 1–11, 2013.

Andrews, S., Tsochantaridis, I., and Hofmann, T. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*. MIT Press, 2003.

Bao, L. and Intille, S. S. Activity recognition from user-annotated acceleration data. In *Pervasive Computing, Second International Conference, PERVASIVE 2004, Vienna, Austria, April 21-23, 2004, Proceedings*, pp. 1–17, 2004.

Bauman, A., Phongsavan, P., Schoeppe, S., and Owen, N. Physical activity measurement–a primer for health promotion. *Promotion and Education*, 13(2):92–103, 2006.

Berlin, E. and Laerhoven, K. Van. Trainspotting: Combining fast features to enable detection on resource-constrained sensing devices. In *The Ninth International Conference on Networked Sensing Systems (INSS 2012)*, pp. 1–8. IEEE Press, 2012.

Cappé, O. Online expectation-maximisation. *Mixtures: Estimation and Applications*, pp. 31–53, 2011.

Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S. T., Trster, G., del R. Milln, J., and Roggen, D. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34:2033 – 2042, 2013.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977. ISSN 00359246.

Deselaers, T. and Ferrari, V. A conditional random field for multiple-instance learning. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 287–294, Madison, WI, 2010. International Machine Learning Society.

Dietterich, T. G. Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pp. 15–30, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44011-9.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, Tomás. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

Foulds, J. and Frank, E. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25: 1–25, 3 2010. ISSN 1469-8005.

Foulds, J. R. and Smyth, P. Multi-instance mixture models and semi-supervised learning. In *SIAM International Conference on Data Mining*, 2011.

Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., and Landay, J. A. Myexperience: A system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, pp. 57–70, 2007.

Guan, X., Raich, R., and Wong, W-K. Multi-instance learning for activity recognition from time series data using a mixture of auto-regressive processes. In *NIPS workshop*, 2015.

Hajimirsadeghi, H., Li, J., Mori, G., Zaki, M., and Sayed, T. Multiple instance learning by discriminative training of markov networks. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, 2013.

Hamilton, J. D. *Time Series Analysis*. Princeton University Press, 1994.

Hu, B., Chen, Y., and Keogh, E. Time series classification under more realistic assumptions. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 578–586, 2013.

Juang, B. and Rabiner, L. Mixture autoregressive hidden markov models for speech signals. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33(6): 14041413, 1985.

Kandemir, M. and Hamprecht, F. A. Instance label prediction by dirichlet process multiple instance learning. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI 2014, Quebec City, Quebec, Canada, July 23-27, 2014*, pp. 380–389, 2014.

Lester, J., Choudhury, T., Kern, N., Borriello, G., and Hannaford, B. A hybrid discriminative/generative approach for modeling human activities. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pp. 766–772, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

Maron, O. *Learning from ambiguity*. PhD thesis, Dept. of EECS, MIT, Cambridge, MA, 1998.

Popescu, M. and Mahnot, A. Early illness recognition using inhome monitoring sensors and multiple instance learning. *Methods of Information in Medicine*, 51(4): 359–367, 2012.

Ravi, N., Dandekar, N., Mysore, P., and Littman, M. L. Activity recognition from accelerometer data. In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, IAAI'05, pp. 1541–1546. AAAI Press, 2005. ISBN 1-57735-236-x.

Stikic, M., Larlus, D., Ebert, S., and Schiele, B. Weakly supervised recognition of daily life activities with wearable sensors. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2521–2537, 2011.

van Kasteren, T., Noulas, A., Englebienne, G., and Kröse, B. Accurate activity recognition in a home setting. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pp. 1–9, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-136-1.

Warrell, J. and Torr, P. H. S. Multiple-instance learning with structured bag models. In *Proceedings of the 8th International Conference on Energy Minimazation Methods in Computer Vision and Pattern Recognition*, pp. 369–384, 2011.

Weidmann, N., Frank, E., and Pfahringer, B. A two-level learning method for generalized multi-instance problems. In *In Proceedings of the Fourteenth European Conference on Machine Learning*, 2003.

Wong, C. S. and Li, W. K. On a mixture autoregressive model. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62(1):pp. 95–115, 2000. ISSN 13697412.

Wu, T-Y., Hsu, J. Y-J., and Chiang, Y-T. Continuous recognition of daily activities from multiple heterogeneous sensors. In *AAAI Spring Symposium: Human Behavior Modeling*, pp. 80–85, 2009.

Ye, L. and Keogh, E. Time series shapelets: A new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pp. 947–956, 2009.

Zhang, D., Liu, Y., Si, L., Zhang, J., and Lawrence, R. D. Multiple instance learning on structured data. In *Advances in Neural Information Processing Systems 24*, pp. 145–153, 2011.

Zheng, Y., Wong, W.-K., Guan, X., and Trost, S. Physical activity recognition from accelerometer data using a multi-scale ensemble method. In *Proceedings of the Twenty-Fifth Annual Conference on Innovative Applications of Artificial Intelligence*, 2013.

Zhou, Z-H., Sun, Y-Y., and Li, Y-F. Multi-instance learning by treating instances as non-i.i.d. samples. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 1249–1256, 2009.

Zhou, Z-H., Zhang, M-L., Huang, S-J., and Li, Y-F. Multi-instance multi-label learning. *Artificial Intelligence*, 176 (1):2291 – 2320, 2012.