Adversarial Unsupervised Representation Learning for Activity Time-Series

Karan Aggarwal,[1] Shafiq Joty,[2] Luis Fernandez-Luque,[3] Jaideep Srivastava[1]

[1]University of Minnesota, [2]Nanyang Technological University, [3]Qatar Computing Research Institute

aggar081@umn.edu, srjoty@ntu.edu.sg, lluque@hbku.edu.qa, srivasta@umn.edu

Abstract

Sufficient physical activity and restful sleep play a major role in the prevention and cure of many chronic conditions. Being able to proactively screen and monitor such chronic conditions would be a big step forward for overall health. The rapid increase in the popularity of wearable devices provides a significant new source, making it possible to track the user's lifestyle real-time. In this paper, we propose a novel unsupervised representation learning technique called *activity2vec* that learns and "summarizes" the discrete-valued activity time-series. It learns the representations with three components: (*i*) the co-occurrence and magnitude of the activity levels in a time-segment, (*ii*) neighboring context of the time-segment, and (*iii*) promoting subject-invariance with adversarial training. We evaluate our method on four disorder prediction tasks using linear classifiers. Empirical evaluation demonstrates that our proposed method scales and performs better than many strong baselines. The adversarial regime helps improve the generalizability of our representations by promoting subject invariant features. We also show that using the representations at the level of a day works the best since human activity is structured in terms of daily routines.

Introduction

Physical activity and sleep are crucial to health and well-being. Requisite activity and sufficient sleep prevent various illnesses such as diabetes (Warburton, Nicol, and Bredin 2006). Rise in chronic conditions, mainly due to aging and unhealthy lifestyles, is putting our healthcare system under stress. The current disorder screening approach requires subjects to undergo various diagnosis steps, involving questionnaires and *polysomnography* (PSG). With increasing popularity of wearable devices like *Fitbit*, which collect detailed data about the body's movements, there is an increased interest in using actigraphy for detecting sleep-related disorders and tracking longitudinal changes in the subject's condition. Although much lower fidelity than clinical devices, availability of wearables provides a novel opportunity, owing to its non-intrusive and real-time capabilities; specifically, if we can develop techniques to extract information from the vast amounts of body-monitoring data. Such techniques could be useful to assist healthcare professionals as well as help monitor behavioral therapy, e.g., exercise.

Only a minuscule proportion of the population has both their clinical data and wearables data available. Thus a purely supervised approach utilizing the wearables-clinical data corpus is sub-optimal since it renders the activity data from a majority of subjects redundant. Hence, any approach towards using activity signals should utilize the unsupervised learning. Second, an important aspect is that information in actigraphy signals depends on the *subjects* and their *environments*, such as their routines and surroundings (Storm, Heller, and Mazzà 2015) along with measurement errors owing to device design. We propose a two-pronged approach. First, a mapping of the temporal relations of the activities from the time-series to a feature space should be learned. Second, this feature space should take into account the subject's environment, and make the representations invariant to the subject and their environment.

We propose a new method that addresses these challenges. Our *activity2vec* method is an unsupervised representation learning model that learns *distributed* representations for activity signals spanning over a time segment (*e.g.,* at a day level) in a subject invariant manner. We use two public data-sets to evaluate our approach against baselines on four disorder prediction tasks. Using a linear classifier (logistic regression), we show our proposed representation learning method outperforms the baseline time-series methods, with day-level representations performing the best. The **linear** classifier with our learned features performs at par with the convolution neural network baseline trained end-to-end on the tasks. We also demonstrate the effectiveness of inducing subject invariant features.

Unlike traditional time-series methods, our feature vectors can be used to boost the performance of the supervised learning models. It has been shown that using pre-trained vectors to initialize the supervised models produces better performance (Hinton et al. 2012). Our method is general enough to be applied to other domains with similar time-series like activity tracking through smart-phones, traffic monitoring, or other sensor data. Specifically, we make the following contributions:

• **Unsupervised scalable embeddings for activity time-series:** To utilize enormous unlabeled human activity data, we propose a novel unsupervised representation learning method *activity2vec* that uncovers activity patterns through *distributed* representation in a scalable fashion, which can
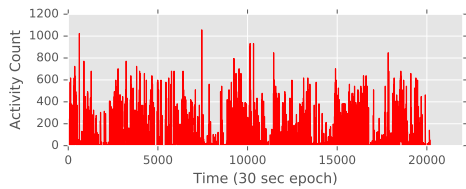
Figure 1: Activity time-series for a subject over a week.

be leveraged towards prediction tasks.

• **A hierarchical model of representation:** One of the persistent challenges in the time-series domain is the selection of time-segments granularity that serve as the primary analysis units. We explore learning representations at various levels of time granularity. We devise a novel algorithm that optimizes two different measures to capture local and global patterns in the activity time-series, along with an ordinal loss to account for the natural ordering in the activities.

• **Subject-invariant representations:** The noise from the subjects environments can hinder generalization. In order to make the representations invariant to subject environments, we train our representations with a subject invariance loss.

Related Work

• **Human activity for health informatics:** Wearable sensors have mostly been used for human activity recognition (HAR) task in machine learning (Bulling, Blanke, and Schiele 2014; Alsheikh et al. 2016), while medical practitioners perform a manual examination on the actigraphy data for diagnosing mostly sleep-disorders (Sadeh 2011). Recent works have tried using actigraphy data for quantifying sleep quality using deep learning (Sathyanarayana et al. 2016) or for actively monitoring human behavioral patterns (Althoff et al. 2017). The novelty of our method is that we propose task-agnostic models rather than plain supervised learning.

• **Representation learning:** Bengio, Courville, and Vincent provide an overview of representation learning, used to construct a space that is discriminative for downstream tasks. It is based on ideas of better network convergence by adding (unsupervised) pre-trained vectors that encode the mutual information between the input features (Goodfellow, Bengio, and Courville 2016). Recently, the area has made enormous progress in NLP, vision, and speech (Collobert et al. 2011; Hinton et al. 2012). Of particular interest are the distributed bag-of-words (DBOW) architectures (Mikolov et al. 2013; Grover and Leskovec 2016; Saha et al. 2017) optimized to predict the context of the structure, unlike continuous-bag-of-words (CBOW) that predicts the structure from its context. In a similar fashion, we use DBOW to capture local patterns in a time segment. Our novelty lies in integrating it in an adversarial setting (Ganin et al. 2016) with an unsupervised predictor, consisting of DBOW with global context and ordinal constraints.

• **Time series analysis literature:** These methods mainly use pair-wise similarity concept to perform classification (Bagnall et al. 2017) and clustering tasks, with

a distance metric like Euclidean. Dynamic Time Warping (Berndt and Clifford 1994) is a widely used technique for finding similarity between two time-series which is computationally expensive due to its pair-wise similarity approach. This has led to the creation of symbolic representation techniques like SAX (Lin et al. 2007) and BOSS (Schäfer 2015), that convert time-series into a symbolic sequence based on amplitudes or frequency analysis, respectively. SAX-VSM and BOSS-VS (Schäfer 2016) use tf-idf (term frequency-inverse document frequency) transformation of these symbolic sequences to get vector representation of sequence windows. HCTSA (Fulcher and Jones 2014) is an *unsupervised* time-series feature extraction engine with over 7800 feature space based on frequency-domain and time-domain analysis of the time-series, unlike the above supervised vector space models. These time-series models, however, cannot complement the supervised learning model, unlike our model's embeddings that can be used to initialize the architectures with back-propagation like neural networks.

Our Approach

In this section, we describe our method *activity2vec*. We first describe challenges, followed by the model components that address these challenges.

Challenges for *activity2vec*

To create a representational schema for activity time-series, the first natural challenge is determining the **right granularity** of the analysis unit. For example, consider the sample time-series in Figure 1, where the x-axis represents the time in 30 seconds epochs and the y-axis represents the activity levels, which in our setting are discrete values ranging from 0 to 5000. Learning representations for each activity level might result in sparse vectors that are too fine-grained to be effective in the downstream tasks. Similarly, learning a representation with too big an analysis unit (*e.g.,* spanning a week) could result in generic vectors lacking required discriminative power. As we demonstrate in our experiments, the right level of granularity is somewhere in between (a day span). Within the analysis units, the **relative magnitude** of the signal values (*e.g.,* '10' < '15') should accounted for.

Considering granularity of analysis unit shorter than the sequence length posits another challenge – how to capture the **global contextual dependencies** between the units. Since the units are parts of a sequence that describes a person's activity over a timespan, they are likely to be interdependent which should be captured in the representations. The same activity can look very different across the subjects owing to **subject-specific** noise and environment, *e.g.,* how they wear the device on their wrist. Our *activity2vec* model addresses these challenges as described in the next section.

Problem Formulation and Time Granularity

Let $S = \{S_1, S_2, \cdots, S_N\}$ denote a set of activity sequences for P unique subjects, where each sequence $S_n = (t_1, t_2, \ldots, t_l)$ is l-length activity (*e.g.,* step counts) for a subject p over a time period (a week here). Let $g \in \{30$ sec, 1 hour, 1 day, 1 week$\}$ specify the granularity we want
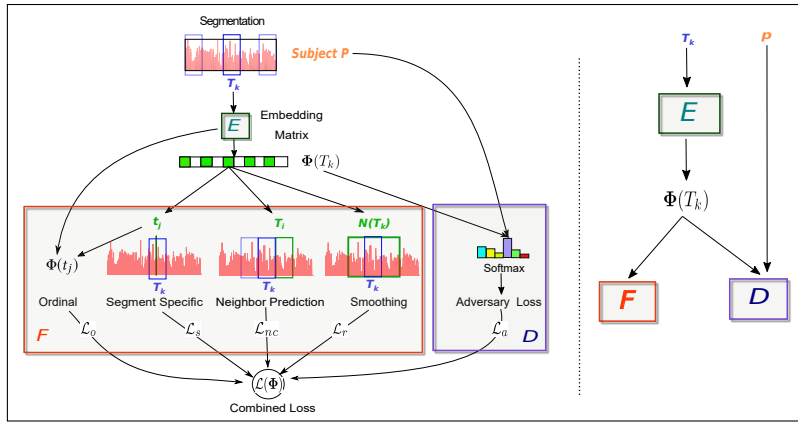
Figure 2: Graphical flow of *activity2vec*'s components: encoder E (embedding matrix), regularized predictors F, and the subject discriminator D. Figure on left shows the sub-losses of the three components, while figure on right shows the overall schema of a three-player game between F, D, and E. First, segmentation is done based on the time granularity. For a selected segment T_k, its embedding $\Phi(T_k)$ is first looked up from E. The embedding is then used by the predictors in F and the discriminator D. The encoder E plays a cooperative game with F to allow it to induce the necessary information. E also plays an adversarial min-max game against D to prevent it from identifying the subject from the embedding to promote subject invariance.

to encode. We first break each sequence S_n into K consecutive time segments of equal length based on g (top of Figure 2). Let $T_k = (t_a, t_{a+1}, \ldots, t_{a+L-1}) \in \mathcal{T}$ be such a segment of length L that starts at time a. Our aim is to learn a mapping function $\Phi : \mathcal{T} \to \mathbb{R}^d$ to represent each time segment T_k by a d dimensional representation. Equivalently, if we represent each time segment in the dataset with a unique identifier (ID), the mapping function Φ is a lookup operation on an embedding matrix of a single hidden layer neural network with non-linear activations. Our goal is to learn the embedding matrix by considering the segment's *content* and *context* while promoting subjective invariance. S_n can be obtained by concatenating (or averaging) the K segment-level vectors; we use concatenation. In this work, we consider the following time spans for a comparative analysis:

- **Sample (`sample2vec`):** representation for each 30-second sample. Our 20,160 length time-series yields a representation space of $\mathbb{R}^{20160 \times d}$.

- **Hour (`hour2vec`):** representation for one-hour chunks, producing a vector space of $\mathbb{R}^{168 \times d}$.

- **Day (`day2vec`):** embeds time-series at the level of a day span, giving us a representational space of $\mathbb{R}^{7 \times d}$.

- **Week (`week2vec`):** provides embeddings at the scale of a week, yielding a vector in $\mathbb{R}^d$ space.

For a given granularity level, *activity2vec* learns the mapping function Φ. While it is possible to use a pre-processing step with change-points detection (CPD) to get the time-segments instead of manually setting the granularity, this step needs careful adjustments to set the thresholds for CPD models adding to further complexity of the model. In principle, we can have a space search over the possible granularities instead of using the pre-defined set above. We skip that step in this work for the sake of simplicity. Instead, we

demonstrate the model's behavior for the choice of granularities that are intuitive to humans. Figure 2 presents the graphical flow of *activity2vec*.

Our model relates to the sequential methods like LSTM by taking into account the global temporal dependencies. Considering the inter-segment and intra-segment context is analogous to a shallow auto-encoder with very dense localized connections and sparse neighboring connections. While we leverage the discrete valued nature of our time-series, we can apply this method to continuous valued time-series by discretization, as done in the traditional time-series literature. The adversarial setting motivated by the environmental noise, which might not apply to other domains. We first present the loss components, and then the combined loss.

Modeling Segment Content in *activity2vec*

We use two loss functions in *activity2vec* to capture the content of a segment – the ordinal relation between time-series values and their co-occurrence patterns.

Segment-Specific Loss We use the segment-specific loss to learn a representation for each time segment by predicting its own values. Given an input sequence $T_k = (t_a, t_{a+1}, \ldots, t_{a+L-1})$, we first map it to a unique vector $\Phi(T_k)$ by looking up the corresponding vector in the shared embedding matrix Φ. We then use $\Phi(T_k)$ to predict each symbol t_j sampled randomly from a window in T_k. However, using a softmax layer for the prediction is very computationally expensive. To compute the prediction loss in an efficient manner, we use Noise-Contrastive Estimation (Gutmann and Hyvärinen 2012) as an alternative to softmax:

$$\mathcal{L}_s(\boldsymbol{\Phi}, \mathbf{W_s} | T_k, t_j) = -\log \sigma(\mathbf{w}_{t_j}^\top \boldsymbol{\Phi}(T_k)) \tag{1}$$
$$-\sum_{m=1}^{M} \mathbb{E}_{t_m \sim \nu(t)} \log \sigma(-\mathbf{w}_{t_m}^\top \boldsymbol{\Phi}(T_k))$$

where σ is the sigmoid function defined as $\sigma(x) = 1/(1 + e^{-x})$, $\mathbf{w}_{t_j}$ and $\mathbf{w}_{t_m}$ are the weight vectors associated with t_j and t_m symbols, respectively, $\nu(t)$ is the noise distribution from which negative example t_m is sampled, and M is the number of negative examples sampled for each positive example. In our experiments, we use unigram distribution as the noise distribution with $M = 12$.

Since we ask the same segment-level vector to predict its symbols, the model captures the overall pattern of a segment. Note that except for `sample2vec`, the model learns embeddings for both segments and symbols. A segment-based approach is commonly used in the time-series analysis, though among the representational models only models like SAX-VSM (Senin and Malinchik 2013) look at the co-occurrence statistics, with a *bag-of-words* assumption.

Ordinal Regression Loss   When predicting an activity symbol, the segment-specific loss above treats each symbol independently. However, since the symbols represent activity levels, there is a natural ordering in their values (*e.g.,* '5' > '1'), which should be preserved in representations. In order to embed this ordinal relation, we use ordinal regression loss while learning the representation for each activity value:

$$\mathcal{L}_o(\mathbf{\Phi}, \theta, \mathbf{w}_o | t_j) = -\sum_{c=1}^{V} \mathbb{I}(t_j = c) \log \left[\sigma(\theta_{\mathbf{c}} - \mathbf{w}_o^\top \Phi(c)) \right.$$
$$\left. - \sigma(\theta_{\mathbf{c-1}} - \mathbf{w}_o^\top \Phi(c)) \right] \quad (2)$$

where $\sigma(x)$ is the sigmoid function as defined before, $\mathbb{I}$ is the indicator function, and V is the number of distinct discrete values that time-series can take (0-5000 in our case). Here, $\sigma(\theta_{\mathbf{c}} - \mathbf{w}_o^\top \Phi(c)) = p(t_j \leq c | \Phi(c))$ is the cumulative probability of t_j being at most c with θ_c being the ordered threshold for the regression, such that $\theta_j > \theta_i, \forall j > i$.
Remark: In principle, we can integrate the ordinal relation in Equation 1 with the NCE loss. However, owing to the hierarchical nature of our algorithm, the segment ID is also predicted while learning the symbols and segments representations. Since these segment-level IDs do not have an ordinal relation with the symbol IDs, we resort to using an ordinal loss applicable only to time-series symbols.

Modeling Segment Context in *activity2vec*

Loss functions presented above capture local patterns in a segment. However, since the segments are contiguous and describe activities of a person, they are likely to be related. For example, after a strenuous activity, there might be lighter activity periods. Likewise, one can expect a smooth transition from one segment to the next. The algorithm should capture relations between proximal segments. With this motivation, we use two loss functions to model this relationship.

Neighbor Context Loss   Similar to activity symbols, each segment in the dataset is assigned a unique identifier that we can use to look up its corresponding vector in the embedding matrix or to predict the segment ID. We first formulate the relation between neighboring segments by asking the current segment vector $\mathbf{\Phi}(T_k)$ (*i.e.,* looked-up vector for segment T_k) to predict its neighboring segments in the time-series: T_{k-1} and T_{k+1}. If T_i is a neighbor to T_k, the neighbor context loss is the neighbor prediction task using NCE as before:

$$\mathcal{L}_{nc}(\mathbf{\Phi}, \mathbf{W_{nc}} | T_k, T_i) = -\log \sigma(\mathbf{w}_{T_i}^\top \mathbf{\Phi}(T_k)) \quad (3)$$
$$-\sum_{m=1}^{M} \mathbb{E}_{T_m \sim \nu(T)} \log \sigma(-\mathbf{w}_{T_m}^\top \mathbf{\Phi}(T_k))$$

where, $\mathbf{w}_{T_i}$ and $\mathbf{w}_{T_m}$ are the weight vectors associated with T_i and T_m segments, respectively. The noise distribution $\nu(T)$ is as described before over segment IDs.

Smoothing Loss   While the previous objective attempts to capture neighborhood information, we also hypothesize that there is a "continuity" between neighboring segments. The learning algorithm should discourage any abrupt changes in the representation of proximal segments. We apply *smoothing between the neighboring segments* by minimizing the l_2-distance between representations of the neighbors:

$$\mathcal{L}_r(\mathbf{\Phi} | T_k, \mathcal{N}(T_k)) = \frac{\eta}{|\mathcal{N}(T_k)|} \sum_{T_c \in \mathcal{N}(T_k)} \|\mathbf{\Phi}(T_k) - \mathbf{\Phi}(T_c)\|^2$$
$$(4)$$

where $\mathcal{N}(T_k)$ is the set of time-segments in proximity to T_k and η is the smoothing strength parameter. Note that the smoothing loss is not applicable to `week2vec`.

Modeling Subject Invariance in *activity2vec*

One challenge in dealing with human activity data is that it heavily depends on the subject's environment. To promote subject invariance, we use an *adversary* loss (Ganin et al. 2016). Let P be the number of unique subjects. We use a multi-class classifier as a *discriminator* to predict the time-segment's source (subject) $s \in \{1, \ldots, P\}$ from the encoded segment representation $\mathbf{\Phi}(T_k)$. In other words, the discriminator tries to identify the subject from whose activity time-series the *encoder* (the embedding matrix) has derived the segment's representation. Note that we want to emphasize the invariance from the source subject of the time-segment and not from the particular sequence from which the segment is derived. Hence, if we have multiple sequences for the same subject, the sequences will share the same subject s in our model. Formally, the discriminator is defined by a soft-max:

$$p(s = p | \Phi(T_k), \mathbf{U}) = \frac{\exp(\mathbf{u}_p^T \mathbf{\Phi}(\mathbf{T_k}))}{\sum_{p'} \exp(\mathbf{u}_{p'}^T \mathbf{\Phi}(\mathbf{T_k}))} \quad (5)$$

where $\mathbf{u}_p$ is the weight vector associated with subject p, and $\mathbf{U}$ defines all the parameters of the discriminator. We use a cross-entropy loss to optimize the subject discriminator:

$$\mathcal{L}_d(\mathbf{U} | \mathbf{\Phi}, T_k, s = p) = -\sum_{s=1}^{P} \mathbb{I}(s = p) \log p(s = p | \Phi(T_k), \mathbf{U}) \quad (6)$$

where p is a subject from whom the segment T_k is encoded. With a goal to promote subject invariance, we put the encoder (the embedding matrix) in adversary with the discriminator. The encoder attempts to encode features that are indistinguishable to the subject discriminator by minimizing (negative of discriminator loss):

$$\mathcal{L}_a(\mathbf{\Phi} | \mathbf{U}, T_k, s = p) = \sum_{s=1}^{P} \mathbb{I}(s = p) \log p(s = p | \Phi(T_k), \mathbf{U}) \quad (7)$$

Combined Loss for *activity2vec*

We define our *activity2vec* model as the combination of the losses described in Equations 1, 2, 3, 4, and 7:

$$\mathcal{L}(\mathbf{\Phi}) = \sum_{n=1}^{N} \sum_{T_k \in S_n} \sum_{\substack{t_j \in T_k \\ T_i \in \mathcal{N}(T_k)}} \Big[\underbrace{\mathcal{L}_s(\mathbf{\Phi}, \mathbf{W}|T_k, t_j) + \beta \mathcal{L}_o(\mathbf{\Phi}, \theta, \mathbf{w}_o|t_j)}_{\text{Segment Content}} +$$

$$\underbrace{\mathcal{L}_{nc}(\mathbf{\Phi}, \mathbf{W}|T_k, T_i) + \mathcal{L}_r(\mathbf{\Phi}|T_k, \mathcal{N}(T_k))}_{\text{Segment Context}} + \underbrace{\lambda \mathcal{L}_a(\mathbf{\Phi}|\mathbf{U}, T_k, s)}_{\text{Adversarial}} \Big]$$

(8)

where $\beta > 0$ and $\lambda > 0$ are the relative weights for the ordinal regression loss and the subject invariance loss, respectively. Concurrently, we also minimize the discriminator loss in Equation 6. As shown at the right in Figure 2, the training of *activity2vec* involves an optimization game between three players: the encoder (E), the combined predictor (F), and the discriminator (D). E plays a cooperative game with F to allow it to induce the necessary information. E also plays an adversarial min-max game against D to prevent it from identifying the subject from the encoded vector to promote subject invariance. We train our model using stochastic gradient descent (SGD).

The main challenge in adversarial training is to balance the two components – the combined loss in Eq. 8 vs. the discriminator loss in Eq. 6, as shown (right) in Figure 2. If one player becomes smarter, its loss to the shared encoder (embedding matrix) overwhelms, and the training fails to converge (Arjovsky, Chintala, and Bottou 2017). In our experiments, the discriminator converges much faster. To stabilize the training, we update the discriminator once every five gradient steps of the algorithm, chosen randomly. Also, we follow the weighting schedule proposed by (Ganin et al. 2016, p. 21), that initializes λ to 0, and then changes it gradually to 1 as training progresses. Through our experiments we demonstrate that the intuitions captured by the components are synergistic since they improve the performance incrementally as the components are added.

Experimental Settings

In this section, we describe experimental settings — datasets, the prediction tasks on which we evaluate the embeddings, the baselines models, and parameter selection.

Datasets

We use Study of Latinos (SOL) (Sorlie et al. 2010) and Multi-Ethnic Study of Atherosclerosis (MESA) (Bild et al. 2002) datasets. SOL has physical activity and clinical data for 1887 subjects, while MESA only has activity data for 2237 subjects. This simulates the scenario where disorder condition labels are available only for a portion of users. These datasets contain activity data (actigraphy) from each subject for a minimum of 7 days measured with wrist-worn Philip's Actiwatch Spectrum. The time-series for each subject is sampled at a rate of 30 seconds. Actigraphy records the mean activity count reported with ZCM (Zero Crossing Mode) per 30 seconds, providing us with a signal that can only take integer values. This makes embedding the input symbols straightforward, without any pre-processing. The

few missing values ($< 1\%$) observed in the data-set were replaced by unknown (UNK) token. The datasets are not skewed with respect to the class distribution of different prediction tasks described next.

Prediction Tasks

We evaluate the effectiveness of the learned embeddings on the following health disorder prediction tasks:

• **Sleep Apnea:** Sleep apnea syndrome is a sleep disorder characterized by reduced respiration during the sleep time. We use the Apnea-Hypopnea Index (AHI) at 3% desaturation level with AHI<5 being characterized as *non-apneaic*, while AHI > 5 indicating a *mild-to-severe-apnea*.

• **Diabetes:** Diabetes (type 2) is the body's insensitivity to insulin, leading to elevated levels of blood sugar. Task is defined as a three-class classification problem, to decide whether a subject is a *non-diabetic*, *pre-diabetic*, or *diabetic*.

• **Hypertension:** Hypertension refers to abnormally high levels of blood pressure, an indicator of stress. Hypertension prediction characterizes a binary classification problem for increased blood pressure.

• **Insomnia:** Insomnia is a sleep disorder characterized by an inability to fall asleep easily, leading to low energy levels during the day. We use a 3-class prediction problem for classifying subjects into *non-*, *pre-* and *insomniac* groups.

Baseline Models

We compare our method with a number of naive baselines and existing systems that use time-series representations:
(a) Majority Class: This baseline always predicts the class that is most frequent in a dataset.
(b) Random: This baseline randomly picks a class label.
(c) SAX VSM: SAX-VSM (Senin and Malinchik 2013) uses SAX, one of the most widely used time-series representation technique.
(d) BOSS: BOSS (Schäfer 2015) is a symbolic representational learning technique that uses Discrete Fourier Transform (DFT) of time-series windows. BOSS creates equal sized bins from histograms of DFT coefficients, which are then assigned representational symbols. Labels are assigned based on the class that gets the highest similarity score using nearest neighbor approach.
(e) BOSSVS: BOSS in Vector Space (Schäfer 2016) is similar to SAX-VSM and creates vector space representation of the time-series from BOSS. BOSS is known to be one of the most accurate methods on standard time-series classification tasks, with BOSS-VS performing marginally lower.
(g) CNN: We use a Convolutional Neural Network (CNN) with Conv-ReLU-AvgPool-BatchNorm-Dropout layers for supervised prediction. We add layers until we get no performance improvement on held-out set.
(h) HCTSA: Highly Comparative Time-Series Analysis (Fulcher and Jones 2014) is a time-series feature extraction engine with over 7800 features extracted with frequency and time domain analysis like Kurtosis. Unlike the time-series baselines above, this is an unsupervised method — most relevant baseline to our method.

Table 1: F_1 and **Speed** relative to **wall clock time** of our `day2vec+Reg+O+A`. O refers to **O**rdinal and A to **A**dversarial.

	Method	Clf.	Sleep Apnea	Diabetes		Insomnia		Hypertension	Speed
			F_1	F_1-macro	F_1-micro	F_1-macro	F_1-micro	F_1	
Supervised	`Majority`	0-R	00.0	21.7	31.9	47.4	25.5	00.0	-
	`Random`		33.9	34.3	31.3	36.2	30.0	33.4	-
	`SAX-VSM`		00.0	38.6	24.3	47.4	25.5	00.0	-
	`BOSS`		17.6	38.9	31.5	49.8	34.9	29.6	-
	`BOSSVS`		11.7	40.1	32.7	47.5	33.1	31.3	-
	`Task-specific`	CNN	41.5	45.2	41.0	50.7	40.1	36.6	2.0x
Unsupervised	`sample2vec`	LR	36.7	40.0	36.7	42.4	35.3	39.4	-
	`hour2vec`	LR	30.0	41.4	33.3	44.6	28.5	24.4	-
	`hour2vec+Reg`	LR	20.5	42.1	32.0	43.5	28.7	23.1	-
	`day2vec`	LR	36.8	40.9	38.0	45.2	35.8	40.3	0.3x
	`day2vec+Reg`	LR	**38.9**	41.8	**39.5**	46.6	**39.7**	**43.4**	0.3x
	`week2vec`	LR	14.5	40.6	34.1	44.2	31.5	18.7	-
	`HCTSA`	LR	20.3	40.0	35.0	46.7	33.7	22.0	8.2x
	`LSTM`	LSTM	32.2	41.4	33.3	46.1	30.4	37.8	10.5x
	`day2vec+Reg+O`	LR	40.5	45.3	40.2	50.9	40.3	43.6	0.4x
	`day2vec+Reg+O+A`	LR	**43.6**	45.8	**42.5**	55.7	**41.4**	**44.1**	**1.0x**

(i) LSTM: We train an LSTM based unsupervised model similar to (Sundermeyer, Schlüter, and Ney 2012) that learns to encode sequences by predicting next time-series value (as a language model in NLP). Next, we use this pre-trained network to initialize the LSTM network that is further trained on the supervised learning tasks.

Variants of *activity2vec* **(a) Unregularized models:** This group of models contain only two NCE loss components from Equation 8: $\mathcal{L}_s$ and $\mathcal{L}_{nc}$. In the Results section, we refer to these models as `sample2vec`, `hour2vec`, `day2vec`, and `week2vec`.
(b) Regularized models: We add smoothing loss $\mathcal{L}_r$ to models in (a). This group includes `hour2vec+Reg` and `day2vec+Reg`. We omit `sample2vec+Reg` since it performed extremely poorly on all the tasks. Recall that smoothing is not applicable to `week2vec`.
(c) Ordinal loss model: We use ordinal loss $\mathcal{L}_o$ with these models. We add this loss to `day2vec+Reg` model in (b), our best performing model as discussed in the next section. We omit other time-unit models for brevity.
(d) Adversarial model: These models use the *adversary* loss $\mathcal{L}_a$ for our `day2vec+Reg` with ordinal loss.

Hyper-Parameter Selection

We use 80%,10%,10% split for train, validation, and test sets repeated 10 times, and we report the mean scores. As mentioned earlier, we only have the disorder task labels for the SOL dataset. For the supervised models we only use the SOL dataset, while for the unsupervised models we use the combined SOL and MESA data. The embedding size of d=100 was fixed for all the models. The weighting parameters λ and β were chosen to be 0.05 and 0.5, respectively. The remaining hyper-parameters in *activity2vec* are: window size (w) for segment-specific loss, number of neighboring segments ($|\mathcal{N}(T_k)|$) and regularization strength (η) for `day2vec` and `hour2vec`. We tuned for $w \in \{12, 20, 30, 50, 100, 120, 500\}$, $\eta \in \{0, 0.25, 0.5, 0.75, 1\}$, and $|\mathcal{N}(T_k)| \in \{2, 4\}$ on the development set. We chose w of size 20, 20, 30, and 50 for `sample2vec`, `hour2vec`, `day2vec`, and `week2vec`, respectively. The η of 0.25 and 0.5 were chosen for `day2vec` and `hour2vec`, respec-

tively. The neighbor set size of 2 was chosen. For the CNN baseline, 3, 4, 3, and 3-layered network were used for sleep-apnea, diabetes, insomnia, and hypertension, with a dropout of 0.5 trained with Adam Optimizer. We tuned all the parameters for maximizing the F_1 scores.

Results and Discussion

In this section, we present our results for the four prediction tasks described in the previous section. The results are presented in Table 1 in four groups: (*i*) baselines, (*ii*) existing time-series representation methods and CNN, (*iii*) our unregularized and regularized *activity2vec* variants, and (*iv*) our ordinal and adversarial *activity2vec* models. We show classification performance in terms of $\mathbf{F}_1$ scores.

• *Performance on disorder classification tasks:* Since our goal is to evaluate the effectiveness of the learned vectors, we use simple **linear classifier** Logistic Regression (LR) with our *activity2vec* models. For the multi-class classification problems like Diabetes and Insomnia, we use One-vs-All classifiers, tuning for micro-F_1 score. We run each experiment 10 times and take the average of the evaluation measures to avoid any randomness in results. We can notice that `day2vec` consistently gives absolute 2-4% improvements over the other *activity2vec* variants, and 6-20% over the baseline time-series representation models and LSTM on F_1 scores on all tasks. The adversarial-ordinal-regularized variant of `day2vec` gives the best results among all the variants. Adversarial variant performs at par or better than task-specific supervised CNN on all the tasks. Due to inherent noisy nature of activity time-series and long sequence length, the LSTM-based model did not work well.

• *Selecting time granularity:* Across the tasks, the adversarial ordinal `day2vec+Reg` outperforms all the models. Models with day level granularity perform better than all the other granularities as well as baseline methods. Among the *activity2vec* variants, the `week2vec` models perform the worst, while `hour2vec` models perform just a bit better on an average. Hour and week-level models perform similarly to the baseline time-series methods. The high-dimensional models based on samples (`sample2vec`) perform better than hour-level, week-level, and baselines. `day2vec` pro-
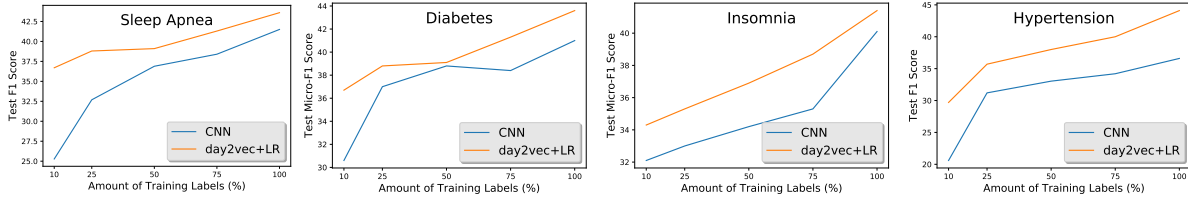
Figure 3: Comparison of our unsupervised `day2vec+Reg+O+A` with LR vs. supervised CNN as a function of labeled data.

duces marginally better results than the `sample2vec` despite much lower dimensional space (2880x). The level of granularity makes a lot of difference in the performance of our models. From the above results, we can conclude that while the low granularity level (`sample2vec`) suffered from coarse embeddings, the high granularity (`week2vec`) level embeddings lost the ability to discriminate.

- **_Effect of smoothing:_** Intuition behind adding the temporal smoothing loss (Eq. 4) to our model was to test the hypothesis that human activities happen in continuity and follow a macro-routine. This should be reflected in neighboring time-segments making them similar in structure. As results suggest, the continuity hypothesis was misguided at the sample- and hour-level. Regularization at the sample level made them lose the discriminative power for classification—considering the noise in activity levels at such a fine granularity. `hour2vec+Reg` exhibits a significant drop (sleep apnea) or at par performance compared to `hour2vec`. Since humans tend to switch between different activity types at the order of hours or less, the hypothesis of continuity was inappropriate at hour level as well. However, adding smoothing helps produce gains for `day2vec`, our best model. We argue that smoothing regularization helps capture a higher level global context since humans structure *routines* at the level of the day, while we switch between activities on the order of hours or lower. This is **supported by the periodogram analysis**, where day level frequencies dominate.

- **_Effect of ordinal loss:_** Addition of ordinal loss improves the accuracy of our models, albeit marginally. While the other loss functions work on co-occurrence of activity values at local and global contexts, ordinal loss explicitly models the relationship between the magnitude of the activity values, *e.g.,* '25' > '5'. While it can be argued that similar ordinal values would co-occur, hence reflected in the $\mathcal{L}_s$, adding

an explicit ordinal constraint helps, though marginally.

- **_Effect of adversarial training:_** Adding the adversarial loss to the ordinal-regularized models improves the F_1 scores across the tasks by 0.5%-3.1% in absolute numbers and 2%-7.5% in relative terms. The difference in performance is more than the 95% confidence intervals around the repeated experimental means reported here. It can thus be concluded that making the representations invariant to the source (subjects) helps in removing the noise introduced by the subjects and their environments, owing to how they wear their device and their routines, making the embeddings more generalized. With an adversarial setting, we can create a representation space more relevant to the health condition of the subjects by removing the subject source domain.

Figure 4 shows the t-SNE (Maaten and Hinton 2008) plot of regularized `day2vec` (left) vs adversarial `day2vec` embeddings (right) for SOL dataset subjects. For each subject, we concatenate the embeddings from constituent day level time-segments. We plot the lifestyles of the subjects as determined by the study questionnaire, identifying each subject as *highly-active*, *moderately-active*, and *sedentary* person. We can notice clusters or subject *phenotypes* with nice separation within the cluster for each of the lifestyle type with non-adversarial `day2vec`. Unsurprisingly, the clusters get very compact in the adversarial setting. The lifestyle classes get clustered markedly separately rather than forming in-cluster separations, observed in the non-adversarial setting. Hence, the adversary helps *activity2vec* with removing the subject-wide variance in the learned embeddings, while still capturing the properties of subject phenotypes. Also, adversary loss improved results on the disorder prediction tasks. Hence, by reducing subject level variance, the adversary loss helps encode a better global representation.

- **Scalability:** Our model (Table 1) takes much less training time compared to the deep unsupervised model like LSTM. In fact, our model takes much less time compared to the supervised CNN that only runs on the labeled data. Since our model has only one hidden layer (*i.e.,* embedding layer) and uses NCE for training, it is **scalable** in practical settings compared to the deep neural models. Additionally, our model offers more **flexibility** to incorporate the alternative intuitions including human knowledge (*e.g.,* the day/hour level representational hierarchy) that would be difficult to do with other deep methods. *Clearly, using more sophisticated deep neural models like deep auto-encoders for a semi-supervised setting would pose scalability challenges.*

Note: We restrict our models to a *transductive* setting. However, it can be easily applied to a *inductive* setting, where
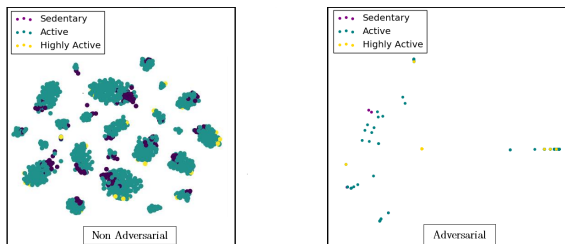

Figure 4: t-SNE visualization of subjects with regularized `day2vec` on the left and adversarial `day2vec` on the right, for all the subjects with respect to their level of activity.

representations for unseen segments can be derived by a single backprop step (Le and Mikolov 2014).

- **Supervised vs Unsupervised:** To show the utility of unsupervised schema, we demonstrate the performance of supervised CNN vs. our adversarial `day2vec` method as a function of the percentage of labeled data in Figure 3. Clearly, our model outperforms CNN across the board. The gap is drastic when the proportion of labeled data is **low**, which is usually the case in practice.

Conclusions

In this work, we present a novel unsupervised representational learning technique, *activity2vec* that encodes human activity time-series by modeling local and global activity patterns. We train our model on two datasets and test on prediction tasks (four commonly occurring disorders). We find that day-level granularity preserves the best representations, which is not surprising since a day is a natural timescale for a full cycle of human activities. Our task-agnostic representational learning model using simple linear classifiers beats baseline time-series representation models on all the disorder prediction tasks. It even performs at par or better than supervised convolutional neural network baseline. Our model learns the representational features using a combination of non-linear loss functions, giving better performance on multiple tasks using simple linear classifiers. We further demonstrate that using adversarial loss along with our embedding encoder model helps increase the performance and generalizability of the embeddings. Additionally, our method is capable of complementing the supervised learning by initialization, unlike existing representation approaches.

References

Alsheikh, M. A.; Selim, A.; Niyato, D.; Doyle, L.; Lin, S.; and Tan, H.-P. 2016. Deep activity recognition models with triaxial accelerometers. In *AAAI Workshops*.

Althoff, T.; Horvitz, E.; White, R. W.; and Zeitzer, J. 2017. Harnessing the web for population-scale physiological sensing: A case study of sleep and performance. In *WWW*, 113–122.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. *CoRR* abs/1701.07875.

Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; and Keogh, E. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *DMKD* 31(3):606–660.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *TPAMI* 35(8):1798–1828.

Berndt, D. J., and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, 359–370. Seattle, WA.

Bild, D. E.; Bluemke, D. A.; Burke, G. L.; Detrano, R.; Diez Roux, A. V.; Folsom, A. R.; Greenland, P.; JacobsJr, D. R.; Kronmal, R.; Liu, K.; et al. 2002. Multi-ethnic study of atherosclerosis: objectives and design. *American journal of epidemiology* 156(9):871–881.

Bulling, A.; Blanke, U.; and Schiele, B. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM CSUR* 46(3):33.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *JMLR* 12(Aug):2493–2537.

Fulcher, B. D., and Jones, N. S. 2014. Highly comparative feature-based time-series classification. *TKDE* 26(12):3026–3037.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *JMLR* 17(1):2096–2030.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *ACM SIGKDD*, 855–864.

Gutmann, M. U., and Hyvärinen, A. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *JMLR 2012* 13(Feb):307–361.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29(6):82–97.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *ICML*, 1188–1196.

Lin, J.; Keogh, E.; Wei, L.; and Lonardi, S. 2007. Experiencing sax: a novel symbolic representation of time series. *DMKD* 15(2):107–144.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *JMLR* 9(Nov):2579–2605.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, 3111–3119.

Sadeh, A. 2011. The role and validity of actigraphy in sleep medicine: an update. *Sleep medicine reviews* 15(4):259–267.

Saha, T. K.; Joty, S.; Hassan, N.; and Hasan, M. A. 2017. Regularized and retrofitted models for learning sentence representation with context. In *CIKM*, 547–556. ACM.

Sathyanarayana, A.; Joty, S.; Fernandez-Luque, L.; Ofli, F.; Srivastava, J.; Elmagarmid, A.; Taheri, S.; and Arora, T. 2016. Impact of physical activity on sleep: A deep learning based exploration. *arXiv preprint arXiv:1607.07034*.

Schäfer, P. 2015. The boss is concerned with time series classification in the presence of noise. *DMKD* 29(6):1505–1530.

Schäfer, P. 2016. Scalable time series classification. *DMKD* 30(5):1273–1298.

Senin, P., and Malinchik, S. 2013. Sax-vsm: Interpretable time series classification using sax and vector space model. In *ICDM*, 1175–1180. IEEE.

Sorlie, P. D.; Avilés-Santa, L. M.; Wassertheil-Smoller, S.; Kaplan, R. C.; Daviglus, M. L.; Giachello, A. L.; Schneiderman, N.; Raij, L.; Talavera, G.; Allison, M.; et al. 2010. Design and implementation of the hispanic community health study/study of latinos. *Annals of epidemiology* 20(8):629–641.

Storm, F. A.; Heller, B. W.; and Mazzà, C. 2015. Step detection and activity recognition accuracy of seven physical activity monitors. *PLoS one* 10(3):e0118723.

Sundermeyer, M.; Schlüter, R.; and Ney, H. 2012. Lstm neural networks for language modeling. In *INTERSPEECH*.

Warburton, D. E.; Nicol, C. W.; and Bredin, S. S. 2006. Health benefits of physical activity: the evidence. *Canadian medical association journal* 174(6):801–809.