Toward Automated Pattern Discovery: Deep Representation Learning with Spatial-Temporal-Networked Data
—Collective, Dynamic, and Structured Analysis

Yanjie Fu
Outline

- **Background and Motivation**
  - Collective Representation Learning
  - Dynamic Representation Learning
  - Structured Representation Learning
  - Conclusions and Future Work
Human-Social-Technologic Systems

IoT, GPS, wireless sensors, mobile Apps

Physical World

Cyber World
Human Activities in Human-Social-Technologic Systems

- **Spatial, Temporal, and Networked (STN) data can be**
  - **Spatial**: Point-of-Interests, blocks, zones, regions
  - **Spatiotemporal**: Taxi trajectories, bus trips, bike traces
  - **Spatiotemporal-networked**: Geo-tagged twitter posts, power grid netload

- **from a variety of sources**
  - **Devices**: phones, WIFIs, network stations, RFID
  - **Vehicles**: bikes, taxicabs, buses, subways, light-rails
  - **Location based services**: geo-tweets (Facebook, Twitter), geo-tagged photos (Flickr), check-ins (Foursquare, Yelp)

Represent the **spatial, temporal, social, and semantic** contexts of **dynamic human/systems behaviors** within and across regions
Important Applications

- User Profiling & Recommendation Systems
- Solar Analytics for Energy Saving
- Intelligent Transportation Systems
- Personalized and Intelligent Education
- Smart Healthcare
- City Governance and Emergency Management
Unprecedented and Unique Complexity

- Spatiotemporally non-i.i.d.
  - Spatial autocorrelation
  - Spatial heterogeneity
  - Sequential asymmetric patterns
  - Temporal periodicity and dependency

Spatial autocorrelations

Sequential asymmetric transitions

Spatial heterogeneity

Temporal periodical patterns

Spatial autocorrelation by distance

Mid-autumn Festival
Unprecedented and Unique Complexity

- Networked over time
  - Collectively-related

- Heterogeneous
  - Multi-source
  - Multi-view
  - Multi-modality

- Semantically-rich
  - Trajectory semantics
  - User semantics
  - Event semantics
  - Region semantics
Technical Pains in Pattern Discovery (1)

- **Feature identification and quantification**
  - Traditional method: Find domain experts to hand-craft features
  - *Can we automate feature/pattern extraction?*
Multi-source unbalanced data fusion

- Traditional method: Extract features, weigh features, weighted combination
- Can we automatically extract features from multi-source unbalanced data?
Field data/real-world systems are usually lack of benchmark labels (i.e., $y$, responses, targets)

- Example: Netload in power grids: behind-the-meter gas-generated electricity and solar-generated electricity are unknown
- Can we learn features without labels (unsupervised)?
Deep Learning Can Help

Task-specific (End to End) Deep Learning

Input

Automated feature learning

Feature extraction + Classification/Clustering

Output

Feature learning from multi-source data

Lack of labels

Generic Deep Learning

Unsupervised Pattern (Feature / Representation) Learning

Classification / Clustering

Car
Not car

Car
Not car
Classic algorithms are not directly available in spatiotemporal networked data

- Traditional method: revised classic algorithms + spatiotemporal networked data regularities
  - Regression + spatial properties = spatial autoregression method
  - Clustering + spatial properties = spatial co-location method
- Can we learn features while maintaining the regularities of spatiotemporal networked data?
Data Regularity-aware Unsupervised Representation Learning

Human and system behaviors have spatiotemporally socially regularities

Generic Deep Learning

Automated feature learning

Feature learning from multi-source data

Data regularities

- Lack of labels (unsupervised)
- Multi-source multi-view multi-modality
- Spatial autocorrelation (peer)
- Spatial heterogeneity (clustering)
- Temporal dependencies (current-past)
- Periodical patterns
- Sequential asymmetric transition
- Spatial hierarchy (hierarchical clustering)
- Hidden semantics
- Spatial locality
- Global and sub structural patterns in behavioral graphs

Data Regularity-aware representation learning

Car
Not car

Lack of labels

Lack of labels (unsupervised)
Automated Feature Learning from Spatial-Temporal-Networked Data

Collective Learning

- Collective representation learning with multi-view data

Dynamic Learning

- Dynamic representation learning with stream data
- Structured representation learning with global and sub structure preservation

Structured Learning
Outline

- Background and Motivation
- Deep Collective Representation Learning
- Deep Dynamic Representation Learning
- Deep Structured Representation Learning
- Conclusion and Future Work
The Rising of Vibrant Communities

- **Consumer City Theory, Edward L. Glaeser (2001), Harvard University.**

- **Spatial Characters:** walkable, dense, compact, diverse, accessible, connected, mixed-use, etc.

- **Socio-economic Characters:** willingness to pay, intensive social interactions, attract talented workers and cutting-edge firms, etc.

What are the underlying driving forces of a vibrant community?

Supported by NSF CISE pre-Career award (III-1755946)
Mobile checkin data

Urban vibrancy is reflected by the frequency and diversity of user activities.

Frequency and diversity of mobile checkins

- Frequency: \( \text{fre} = \#(\text{checkin}) \)
- Diversity: \( \text{div} = - \sum_{\text{type}} \frac{\#(\text{checkin, type})}{\#(\text{checkin})} \log \frac{\#(\text{checkin, type})}{\#(\text{checkin})} \), where \text{type} denotes the activity type of mobile users

Fused scoring

- \( \text{Vibrancy} = (1 + \beta^2) \frac{\text{fre} \times \text{div}}{\beta^2 \times \text{fre} + \text{div}} \)
- \( \beta \) controls the weights of fre and div
- Power-law distributed
- Some are highly vibrant while most are somewhat vibrant
Spatial Unbalance of Urban Community Vibrancy
Motivation Application: How to Quantify Spatial Configurations and Social Interactions

Urban Community = Spatial Configuration + Social Interactions

Static Element

Dynamic Element

Land use in Oxpens

- Civic
- Office/commercial
- Education
- Retail
- Monument/ecclesiastic
- Leisure
- Residential
- Light industrial
- Parking
- Public Access
- Open Space
- Water
From Regions to Graphs

Spatial Regions as Human Mobility Graphs

- POIs $\rightarrow$ nodes
- Human mobility connectivity between two POIs $\rightarrow$ edge weights
- Edge weights are asymmetric
Periodicity of Human Mobility

- Different days-hours → different periodic mobility patterns → different graph structures
Collective Representation Learning with Multi-view Graphs

Spatial Objects (e.g., Regions)

Multiple Graphs

Feature Vector Representations

Constraint: the multi-view graphs are collaboratively related
The encoding-decoding representation learning paradigm

- Encoder: compress a graph into a latent feature vector
- Decoder: reconstruct the graph based on the latent feature vector
- Objective: minimizing the difference between original and reconstructed graphs

- Unsupervised (label-free): doesn’t require labels
- Generic: not specific for single application
- Intuitive: a good representation can be used to reconstruct original signals
Solving Multi-graph Inputs: An Ensemble-Encoding Dissemble-Decoding Method

NN as an input unit of encoder

Minimize reconstruction loss

NN as an output unit of decoder

signal ensemble (Multi-perceptron summation)

signal dissemble (Multi-perceptron filtering)
Solving the Optimization Problem

1. Multi-graph Ensemble Encoding
\[
\begin{aligned}
&y_{i,t}^{(k),1} = \sigma(W_{i,t}^{(k),1} p_{i,t}^{(k)} + b_{i,t}^{(k),1}), \forall t \in \{1, 2, \ldots, 7\}, \\
&y_{i,t}^{(k),r} = \sigma(W_{i,t}^{(k),r} p_{i,t}^{(k)} + b_{i,t}^{(k),r}), \forall r \in \{2, 3, \ldots, o\}, \\
&y_{i,t}^{(k),o+1} = \sigma(\sum_t W_t^{(k),o+1} y_{i,t}^{(k),o} + b_t^{(k),o+1}), \\
&z_i^{(k)} = \sigma(W^{(k),o+2} y_{i,t}^{(k),o+1} + b^{(k),o+2}), \\
\end{aligned}
\]

2. Multi-graph Dissemble Decoding
\[
\begin{aligned}
&\hat{y}_{i,t}^{(k),o+1} = \sigma(\hat{W}_{i,t}^{(k),o+2} z_i^{(k)} + \hat{b}_{i,t}^{(k),o+2}), \\
&\hat{y}_{i,t}^{(k),o} = \sigma(\hat{W}_{i,t}^{(k),o+1} y_{i,t}^{(k),o+1} + \hat{b}_{i,t}^{(k),o+1}), \\
&\hat{y}_{i,t}^{(k),r-1} = \sigma(\hat{W}_{i,t}^{(k),r} \hat{y}_{i,t}^{(k),r} + \hat{b}_{i,t}^{(k),r}), \forall r \in \{2, 3, \ldots, o\}, \\
&\hat{p}_{i,t}^{(k)} = \sigma(\hat{W}_{i,t}^{(k),1} \hat{y}_{i,t}^{(k),1} + \hat{b}_{i,t}^{(k),1}), \\
\end{aligned}
\]

3. Objective Function
\[
L^{(k)} = \sum_{t \in \{1, 2, \ldots, 7\}} \sum_i \| (p_{i,t}^{(k)} - \hat{p}_{i,t}^{(k)}) \odot v_{i,t}^{(k)} \|_2^2
\]

Sparsity regularization: If mobility connectivity = 0, weight=1 to penalize the loss
If mobility connectivity >0, weight>1
Comparisons with Features Generated By Different Methods

- **Data**
  - Beijing Checkin Data

- **Ranking Models**
  - MART: it is a boosted tree ranking model
  - RankBoost (RB): it is a boosted pairwise ranking method, which trains multiple weak rankers and combines their outputs as final ranking.
  - RankNet (RN): it uses a neural network to model the underlying probabilistic cost function.

- **Feature Sets**
  - Explicit Features (EF)
  - Latent features (LF)
  - Explicit & Latent features (ELF)
  - Features generated by variation 1 of our method: distance graphs not mobility graphs
  - Features generated by variation 2 of our method: average not collective
  - Features generated by variation 3 of our method: non-weighted not unsupervised weighted.

- **Evaluation Criteria**
  - NDCG: Evaluate the ranking performance at Top N
Comparison with Baseline Representation Learning Algorithms

- Ranking Models
  - LAMBDA MART
  - ListNet
  - MART
  - RankBoost

- Baseline Methods
  - RBM: restricted Boltzmann machine
  - NMF: non-negative matrix factorization
  - Skip-gram

- Evaluation Criteria
  - NDCG: Evaluate the ranking performance at Top N
Summary

- **Task**
  - Collective representation learning with multi-view graphs

- **Modeling**
  - Develop an ensemble-dissemble encoding-decoding approach
  - multi-graph ensemble encoding and multi-graph dissemble decoding

- **Application**
  - Quantifying urban communities for understanding urban vibrancy
Outline

- Background and Motivation
- Collective Representation Learning
- **Dynamic Representation Learning**
- Structured Representation Learning
- Conclusion and Future Work
What can we do to defend social fairness on insurance rates?

Washington, D.C. – Many good drivers pay higher insurance premiums because of their credit history and other factors that have nothing to do with their driving record, according to Consumers Union, the policy and advocacy division of Consumer Reports.

The consumer group urged regulators to ban the use of credit histories and some other non-driving factors for setting premiums at a National Association of Insurance Commissioners (NAIC) hearing on November 19th.
Motivation Application: Machine-Learning Based Driving Behavior Analysis

Driving Behavior Analysis

- Turn Left
- Accelerate
- Turn Right

Insurance Companies

- Idle Speed Alert
- Over Speed Beep Alert
- Rapid Acceleration Beep Alert
- Hard Braking Beep Alert
Defining Driving Operations & States

- **Driving Operations**
  - Speed-related:
    - acceleration, deceleration, constant speed
  - Direction-related:
    - Turning right, left, moving straight

- **Driving States**
  - Definition: speed operation + direction operation

<table>
<thead>
<tr>
<th>Speed Operation</th>
<th>Direction Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration</td>
<td>Turning Right</td>
</tr>
<tr>
<td>Deceleration</td>
<td>Turning Left</td>
</tr>
<tr>
<td>Constant Speed</td>
<td>Moving Straight</td>
</tr>
</tbody>
</table>
Quantifying Driving Habits with Driving State Transition Graphs

Left turn + deceleration
Right turn + deceleration
Left turn + deceleration
Straight + Acceleration

Driving style & habit patterns

Transition Duration View
Driving State
Transition Duration

Transition Frequency View
Driving State
Transition Probability
Driving State Transition Graph Sequence

- Transition frequency: how frequently a driver changes his/her driving state from one to another (unusual high-frequency: drunk?)
- Transition duration: how quickly a driver changes his/her driving state from one to another (unusually fast: non-comfortable driving habits)
Dynamic Representation Learning with Graph Stream

\[ f( ) = \begin{cases} \text{Map a sequence of time-varying yet relational graphs to a} \\ \text{sequence of time-varying yet relational vectors} \\ \text{s. t. spatial and temporal dependencies} \end{cases} \]
Three Modeling Constraints

- **Structural Reservation**
  - If two graphs’ structures are similar, their feature vectors are similar

- **Temporal Dependency**
  - Current driving operations are related to previous driving operations

- **Peer Dependency**
  - Drivers with similar driving behaviors should share similar feature vectors
Structural Reservation: Minimizing reconstruction loss

\[
\begin{align*}
    y_1^i &= \sigma(W_1^1 x_i + b_1^1), \\
    y_k^i &= \sigma(W_k^k y_{k-1}^i + b_k^k), \quad \forall k \in \{2, 3, \ldots, o\}, \\
    z_i &= \sigma(W_o^o y_o^i + b_o^{o+1}).
\end{align*}
\]

\[
\begin{align*}
    \hat{y}_i^o &= \sigma(\hat{W}_o^{o+1} z_i + \hat{b}_o^{o+1}), \\
    \hat{y}_i^{k-1} &= \sigma(\hat{W}_k^k \hat{y}_i^k + \hat{b}_k^k), \quad \forall k \in \{2, 3, \ldots, o\}, \\
    \hat{x}_i &= \sigma(\hat{W}_1^1 \hat{y}_i^1 + \hat{b}_1^1).
\end{align*}
\]

The encoding phrase: encode input vector into embedding;
The decoding phrase: decode the embedding to recover input.
Temporal Dependency: Current driving operations are related to previous driving operations.

### Sequential Encode Step

\[
\begin{align*}
(y^1_i)_{\tau} &= \sigma(W^1 x^\tau_i + b^1), \\
(y^k_i)_{\tau} &= \sigma(W^k (y^{k-1}_i)_{\tau} + b^k), \forall k \in \{2, 3, \cdots, o\}, \\
z^\tau_i &= (1 - c^\tau)z^{\tau-1}_i + c^\tau \tilde{z}^\tau_i.
\end{align*}
\]

### Sequential Decode Step

\[
\begin{align*}
(\hat{y}^o_i)_{\tau} &= \sigma(\hat{W}^{o+1} z^\tau_i + \hat{b}^{o+1}), \\
(\hat{y}^{k-1}_i)_{\tau} &= \sigma(\hat{W}^k (\hat{y}^k_i)_{\tau} + \hat{b}^k), \forall k \in \{2, 3, \cdots, o\}, \\
\hat{x}^\tau_i &= \sigma(\hat{W}^1 (\hat{y}^1_i)_{\tau} + \hat{b}^1).
\end{align*}
\]

Feed the output of Autoencoder’s hidden layer into Gated Recurrent Unit.
Peer Dependency: Drivers with similar driving behaviors should share similar latent representations.

Graphical regularization: if a spatial item $i$ and a spatial item $j$ are similar at time $T$, the representation $Z_i$ and $Z_j$ are similar; punished otherwise.

\[
H_c(G^T) = \sum_{u_i \in U} \sum_{u_j \in U, u_i \neq u_j} s_{i,j}^T \cdot ||z_i^T - z_j^T||_2^2
\]

The similarity of driving behavior between the driver $u_i$ and $u_j$ at the time slot $\tau$

\[
s_{i,j}^\tau = \cos(x_i^\tau, x_j^\tau)
\]

using descriptive statistics of various historical driving operations.
A Joint Optimization Objective

Model Structure

Multi-spatial Graphs Series

\[ G_1 \quad G_2 \quad G_3 \quad \ldots \quad G_m \]

\[ X_1 \quad X_2 \quad X_3 \quad X_m \]

\[
\min \frac{1}{2} \sum_{\tau \in \mathcal{T}} \sum_{u_i \in \mathcal{U}(n)} \| (x_i^{\tau} - \hat{x}_i^{\tau}) \|_2^2 + \alpha \cdot \mathcal{H}_c(G^{\tau})
\]

Structural reservation: the representation that is encoded from input can be decoded to recover input

Temporal dependency: current embedding is related to past embedding

Peer dependency: the similar graph streams from two similar drivers share similar representations
1. Learn driving behavior profiles from driving state transition graphs
2. Use driving behavior profiles to automatically score driving performances and detect risky areas
Comparison with Baseline Methods

Apply the learned representations to predict driving scores

- Our model achieves the best performances
- Peer and temporal dependencies are essential for representing driving behavior

Data
- T-drive (Beijing GPS trajectories of volunteer drivers)

Evaluation Metrics
- Square Error
- Coefficient of Determination ($R^2$)
- Normalized Discounted Cumulative Gain (NDCG@N)
- Kendall Tau Coefficient (Tau)

Baselines
- Autoencoder
- DeepWalk: use truncated random walks to learn latent representations
- LINE: preserve both local and global network structures with an edge-sampling algorithm
- CNN: Convolutional Neural Network
- Driving State Vector (DSV) – a traditional transportation approach
- PTARL—Our model
Study of Peer and Temporal Dependencies

PTARL: -Our model

- Two variants of our model
  - PTARL-peer that only considers the peer dependency.
  - PTARL-temporal that only considers the temporal dependency.

- The Autoencoder that ignores both dependencies performs the worst
- The temporal dependency is more significant in profiling driving behavior than the peer dependency.
A “Riskier Driver” is not always risky

Scores of the “Safer Driver” are relatively higher at most time, while the scores of the “Riskier Driver” are relatively lower at most time.

A “Safer Driver” is not always safe
Risky Area Detection

Dynamic evolution of the distribution of risky areas in 12 hours
Summary

- **Task**
  - Dynamic representation learning with **graph streams**

- **Modeling**
  - Develop a **temporal and peer-aware dynamic representation learning approach**
  - **Robustness checks** over structural preservation, temporal dependency, and peer dependency

- **Application**
  - Driving behavior analysis for inferring driving scores and risk area detection
Outline

- Background and Motivation
- Deep Collective Representation Learning
- Deep Dynamic Representation Learning
- **Deep Structured Representation Learning**
- Conclusion and Future Work
Less Matches Between Human and Technologies

Non-personalized news feeds

What can we do to improve user performance and engagement in human-technological systems?
Webpage = Contents + Structure

User = Explicit Activities + Latent Behavioral Structure?
From Users To Activity Graphs

Spatial-temporal asymmetric transition patterns

User Activity Graph

0.2*0.1 = 0.02
gas station

1.0*0.7 = 0.7
restaurant

0.0*0.1 = 0.0
market

0.2*0.1 = 0.02
home
Given a user and corresponding user activity graph, we aim to map the user to a profile vector.
Global structures: how a user’s activities globally interact with each other (strongly link, weakly link, no link)
Substructure Behavioral Patterns

- Substructures: topology of subgraphs that feature the unique behavioral patterns of a user’s activities

Substructure 1: high-frequency discrete nodes

Substructure 2: high-frequency circle
Traditional solution: global structure (encoding-decoding) + substructure (loss regularization)

- **Global structure:**
  - Minimize the loss between the input graph and the reconstructed graph
- **Substructure preservation:**
  - Strongly penalize the loss if the model cannot accurately reconstruct substructures
Will The Traditional Solution Work?

Different users show different substructure topology and contents

Substructure are dynamically distributed in different locations of graphs
Translate substructure-aware representation learning into an adversarial substructured learning problem

An encoder-decoder network: learn the representations of a graph

A substructure detector: detect substructure patterns

A discriminator: classify original substructure and reconstructed substructure

Adversarial training: to match original substructures with reconstructed substructures
Will The New Formulation Work?

- Traditional subgraph detection algorithms are usually not differentiable
- Impossible to backpropagate gradient for optimization
How to Approximate Substructure Detector?

Non-differentiable substructure detection algorithm

Differentiable CNN

• Use CNN to replace substructure detection algorithms
• Use an embedding vector to replace a subgraph
Approximated Adversarial Substructured Learning

The Mini-Max Game in Optimization

- Discriminator: is trained to maximize the accuracy of classifying detected and generated substructures
- Generator: is trained to minimizing the probability that Discriminator correctly classify generated substructures

CNN-based detector: detect and output a substructure feature vector
Solving The Min-Max Game

1. Minimizing Objective Function

\[ \mathcal{L} = -\lambda_D \mathcal{L}_D + \lambda_G \mathcal{L}_G + \lambda_{AE} \mathcal{L}_{AE} \]

Train G to minimize D’s accuracy on generated substructures

2. Update discriminator to maximize accuracy

\[ -\nabla \theta_d \frac{1}{m} \sum_{i=1}^{m} [\log D(s^i) + \log(1 - D(G(x^i)))] \]

Maximize likelihood
Classify ground true substructure to 1
Classify generated substructure to 0

3. Update generator to confuse discriminator

\[ \nabla \theta_{AE} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(x^i))) \]

Minimize likelihood
Classify generated substructure to 0

4. Minimize reconstruction loss

\[ \nabla \theta_{AE} \| (x_i - \hat{x}_i) \|_2^2 \]
Recap: Training and Testing of Adversarial Substructured Learning

(a) Pre-train the CNN to approximate the substructure detector.

(b) Adversarial training process to integrate the substructure.

(c) Utilize the well-trained model to generate representations of mobile user profiles.
What To Do Next: Inferring Next Activity for POI Recommendations

1. Given a time period, learn a user’s profiles from corresponding user activity graph
2. Exploit user profiles to forecast next activity category
Performance Comparisons on New York and Tokyo Activity Checkin Data

Apply the learned representations to predict next activity type (next POI category)

- **Data**
  - Mobile activity checkin data of NYC and Tokyo

<table>
<thead>
<tr>
<th>City</th>
<th># Check-ins</th>
<th># POI Categories</th>
<th>Time Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>227428</td>
<td>251</td>
<td>12 April 2012 to 16 February 2013</td>
</tr>
<tr>
<td>Tokyo</td>
<td>573703</td>
<td>247</td>
<td>12 April 2012 to 16 February 2013</td>
</tr>
</tbody>
</table>

- **Evaluation Metrics**
  - The precision@N of activity category prediction
  - The precision@N of new activity recommendation

- **Baselines**
  - Autoencoder
  - DeepWalk: use truncated random walks to learn latent representations
  - LINE: preserve both local and global network structures with an edge-sampling algorithm
  - CNN: Convolutional Neural Network

- Our model achieves the best performances on user profiling
- Substructures in a graph are essential for user behavior patterns
Study of Node and Circle Substructures

- StructRL: consider both nodes and circles
- StructRL-Node: only consider node based substructure
- StructRL-Circle: only consider circle based substructure
Summary

- **Task**
  - Structured representation learning with global and substructure preservation

- **Modeling**
  - Develop an adversarial substructured learning approach
  - Preserving global and substructures via solving the mini-max game

- **Application**
  - Precision user profiling and quantification for personalization and recommender systems