A Survey on Transfer Learning for Urban Spatio-temporal Machine Learning

Yilun JIN

Supervisors: Professor Qiang YANG & Professor Kai CHEN
Committee: Professor Wilfred NG (Chair) & Professor Dit-Yan YEUNG
August 18th, 2021
Outline

• Introduction
• Urban Spatio-temporal Machine Learning
• Transfer Learning
• Urban Spatio-temporal Transfer Learning
  • Spatial Transfer Learning
  • Temporal Transfer Learning
  • Cross-modal Transfer Learning
• Conclusions and Future Work
Outline

• Introduction
• Urban Spatio-temporal Machine Learning
• Transfer Learning
• Urban Spatio-temporal Transfer Learning
  • Spatial Transfer Learning
  • Temporal Transfer Learning
  • Cross-modal Transfer Learning
• Conclusions and Future Work
Cities and Urban Problems

• Urbanization leads to many large cities
  • 54% population live in cities [UN, 2015]
  • Metropolises: New York, London, Tokyo, Hong Kong, etc.

• ...and also leads to important urban problems.
  • Environmental pollution
  • Energy consumption
  • Traffic management
  • ......
Machine Learning for Urban Problems

- **Triggers**: Sensing and machine learning
  - Smart sensors, GPS, etc. generate spatio-temporal urban data.
    - E.g. Trajectories, social networks, environment sensors...
  - Machine learning models capture spatio-temporal information.
    - Spatio-temporal machine learning (STML): Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Graph Neural Networks (GNN), ...

- **Applications**:
  - **Traffic**: Speed, travel time, ...
  - **Human activities**: Human flow, events,
  - **Businesses**: Demand & supply, recommendation, ...

Traffic jams [Zhang et al. 2020]

Human Flow [Liang et al. 2021]
Data scarcity in cities

- Machine learning relies on big data.
  - More data leads to better performance.
- Data scarcity is common in cities.
  - Building new cities
  - Planning new urban services
- Problem: How to mitigate lack of data for urban STML?

Object detection, Performance VS data size
[Sun et al. 2017]

New city: Xiong’an New Area, China

New urban service: Tuen Ma Line, 2021
Solution: Transfer Learning

• **Transfer learning:**
  • **Key idea:** Borrow knowledge from **different but related** tasks.
  • Effective in visual recognition, text mining, etc.
    • Pre-training & fine-tuning in computer vision
    • Cross-domain sentiment classification [Li, 2020].

• **Q:** How can transfer learning be applied to urban STML?

![ImageNet Pre training in CV [Shen et al. 2019]](image1)

![Cross-domain language analysis [Li, 2020]](image2)
Outline

• Introduction
• Urban Spatio-temporal Machine Learning
• Transfer Learning
• Urban Spatio-temporal Transfer Learning
  • Spatial Transfer Learning
  • Temporal Transfer Learning
  • Cross-modal Transfer Learning
• Conclusions and Future Work
Urban Spatio-temporal Machine Learning

Data -> Process & Feed into Models -> Optimize Problems

Q: How is spatio-temporal information used to organize the data?

Q: How to extract the spatio-temporal information within the data?

Q: How to formulate urban computing tasks into machine learning problems?
Urban Spatio-temporal Machine Learning: Data

• Data types in urban STML: Classified by spatio-temporal correlations

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Point</th>
<th>Sequence</th>
<th>Static Map</th>
<th>Static Graph</th>
<th>ST Map</th>
<th>ST Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal Variant?</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Spatial Variant?</td>
<td>No</td>
<td>Optional</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Euclidean</td>
<td>/</td>
<td>Optional</td>
<td>Yes</td>
<td>Optional</td>
<td>Yes</td>
<td>Optional</td>
</tr>
<tr>
<td>Spatial Corr.?</td>
<td>/</td>
<td>Optional</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Non-Euclidean</td>
<td>/</td>
<td>Optional</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Spatial Corr.</td>
<td>/</td>
<td>Optional</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- **Sequence:** Trajectories [Wang et al. 2019a]
- **ST Maps:** Urban Flow [Zhang et al. 2017]
- **Graph:** Traffic network [Yu et al. 2018]
Urban Spatio-temporal Machine Learning: Data

- **Multi-modal** urban data:
  - e.g. Transport, environment, business [Zheng et al. 2014], etc.
- Leveraging multi-modal data is common in urban STML.
  - e.g. Use *road maps, points-of-interests* (POI), *weather*, and *transport* data to predict air quality [Wei et al. 2016a].

![Multi-modal urban data](image1.png)

Multi-modal urban data [Zheng et al. 2014]

Use of multi-modal data [Wei et al. 2016a]
Urban Spatio-temporal Machine Learning: Problems

• Supervised learning: \[
\min_{f, \mathcal{X} \rightarrow \mathcal{Y}} \sum_{i=1}^{D} l(f(X_i), y_i),
\]
  • **Independent** samples, label supervision.
  • Minimize loss function \( l(\hat{y}, y) \)
    • Squared loss \( l(\hat{y}, y) = \|\hat{y} - y\|^2 \) for regression
    • Cross-entropy loss for classification \( l(\hat{y}, y) = -\sum_i y_i \log \hat{y}_i \)

• Examples:
  • **Forecasting**: \( X = (X_{t-k}, \ldots, X_{t-1}), y = X_t \)
  • **Estimation (e.g. Travel Time)**: \( X = (\text{trajectory, metadata}), y = \text{travel time} \)

Urban flow forecasting
[Zhang et al. 2017]
Urban Spatio-temporal Machine Learning: Models

• Models capture spatio-temporal relations.

• **Spatial** relations:
  • Convolutional Neural Nets (**CNN**): **Euclidean**, grid data.
    • Convolutions and pooling capture relations within a k\times k grid.
    • Applications: Region flow forecasting [Zhang et al. 2017]

• Graph Neural Nets (**GNN**): **Non-Euclidean**, network data
  • Aggregate information from irregular neighbors via edges.
  • Applications: Traffic forecasting [Geng et al. 2019]

CNN for region flow forecasting [Zhang et al. 2017]

GNN for taxi demand forecasting [Geng et al. 2019]
Urban Spatio-temporal Machine Learning: Models

- **Temporal** correlations:
  - Recurrent Neural Nets (**RNN**)  
    - Inputs share a **memory** to remember previous observations.

- **Hybrid Models**
  - Jointly model spatio-temporal correlations.
  - Combine RNN with CNN/GNN.
  - e.g. **DCRNN** [Li et al. 2018] for traffic forecasting
  - **ConvLSTM** [Shi et al. 2015] for precipitation.
Outline

• Introduction
• Urban Spatio-temporal Machine Learning
• Transfer Learning
• Urban Spatio-temporal Transfer Learning
  • Spatial Transfer Learning
  • Temporal Transfer Learning
  • Cross-modal Transfer Learning
• Conclusions and Future Work
Transfer learning

• **Definitions** [Pan et al. 2009]:
  • **Domain**: $\mathcal{D} = (\mathcal{X}, P(X))$
    Feature space and feature distribution.
  • **Task**: $\mathcal{T} = (\mathcal{Y}, P(y|X))$
    Label space and label conditional distribution.
  • **Transfer learning**: Improve learning on $\mathcal{D}_T, \mathcal{T}_T$ using knowledge from $\mathcal{D}_S, \mathcal{T}_S$
    $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$

• **Key Challenge**: Identify **domain-invariant** knowledge.

• **Categorization**:
  • Homogeneous: $\mathcal{X}_S = \mathcal{X}_T, \mathcal{Y}_S = \mathcal{Y}_T$
  • Heterogeneous: $\mathcal{X}_S \neq \mathcal{X}_T$ or $\mathcal{Y}_S \neq \mathcal{Y}_T$
Transfer learning: Methods

- Transfer learning methods: “What to transfer”
  - Instance-based: Reuse source instances to train on the target domain.
  - Feature-based: Learn domain-invariant features for both domains and learn a common downstream model.
  - Model-based: Encode knowledge in model parameters and reuse parameters for the target domain.

<table>
<thead>
<tr>
<th>Transfer Methodology</th>
<th>Key Assumption</th>
<th>Key Challenge</th>
<th>Categorization</th>
<th>Related Papers</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance-based</td>
<td>Some source samples are similar to target and can be reused.</td>
<td>Adequately re-weight source samples to train target model.</td>
<td>Non-inductive Inductive</td>
<td>[35, 36, 37][38, 16]</td>
<td>MMD is a common distance metric.</td>
</tr>
<tr>
<td>Feature-based</td>
<td>Source and target data share common latent factors.</td>
<td>Measure and maximize domain invariance of features.</td>
<td>Explicit (domain distance) Implicit (adversarial)</td>
<td>[39, 40, 41, 42][43, 44, 45, 46]</td>
<td></td>
</tr>
</tbody>
</table>
Transfer learning: Instance-based

• **Assumption**: Some samples from source are similar to target.

• **Key Challenge**: Assign weights to source samples to train target model.

• **Representative Method**: TrAdaBoost [Dai et al. 2007].
  - **Intuition**:
    - Wrong **target** samples: Increase weight
    - Wrong **source** samples: Decrease weight (Likely dissimilar from target).
  - **Method**:
    - Source and target samples: $D_S = \{X_i, y_i\}_{i=1}^n, D_T = \{X_j, y_j\}_{j=n+1}^{n+m}, y \in \{0, 1\}$
    - Sample weights $w^0 = [w_1^0, \ldots, w_{n+m}^0]$.
    - Iterative Reweighting:
      - Source
        $$w_{i}^{t+1} = \begin{cases} 
        w_{i}^t \beta |h_t(X_i) - y_i|, & i \leq n, \\
        w_{i}^t \beta_t |h_t(X_i) - y_i|, & n + 1 \leq i \leq n + m.
        \end{cases}$$
      - Target
        $$0 < \beta, \beta_t < 1.$$
Transfer learning: Feature-based

• **Assumption**: Domains share common latent factors.
• **Challenge**: Measure & minimize domain distance to identify common factors.
• **Methods**: Minimize Maximum Mean Discrepancy [Borgwardt et al. 2006]

\[
\text{MMD} (D_S, D_T) = \left\| \frac{1}{|D_S|} \sum_{i=1}^{|D_S|} \Phi_K (x_i^S) - \frac{1}{|D_T|} \sum_{j=1}^{|D_T|} \Phi_K (x_j^T) \right\|_{\mathcal{H}}
\]

\( K \): Kernel function
\( \Phi_K \): Kernel mapping of \( K \)
\( \mathcal{H} \): RKHS of kernel \( K \)

• E.g. Transfer Component Analysis (TCA) [Pan et al. 2010]

\[
\min_{W} \text{tr}(W^T K L K W) + \mu \text{tr}(W^T W) \\
\text{s.t. } W^T K H K W = I
\]

Reformulation of MMD\(^2\)
Transfer learning: Feature-based (Cont)

**Methods: Minimize MMD**
- E.g. DDC [Tzeng et al. 2014]
  - Apply MMD regularization on the output features
  - Jointly minimize classification loss and MMD.

**Methods: Adversarial learning**
- Intuition: Domain invariant features should be
  1. **Discriminative** w.r.t. labels
  2. **Indiscriminative** w.r.t. source VS target domain
- DANN [Ganin et al. 2016]
  - Feature Extractor $G$, Classifier $C$, Domain classifier $D$.
  - Minimize label loss $l_y(G, C)$: Intuition 1
  - Maximize domain loss $l_d(G, D)$: Intuition 2

$$V(G, C, D) = \frac{1}{n_s} \sum_{i=1}^{n_s} l_y^i(G, C) - \lambda \left( \frac{1}{n_s} \sum_{i=1}^{n_s} l_d^i(G, D) + \frac{1}{n_t} \sum_{i=1}^{n_t} l_d^i(G, D) \right),$$
Transfer learning: Model-based

- **Assumption:** Model parameters encode general data structures.
- **Key Challenge:** Find transferrable parameters.
- **Methods:**
  - Train transferrable parameters with regularization
    - e.g. MT-SVM [Evgeniou and Pontil, 2004]
  - Identify transferrable parameters from **well-trained models**
    - [Yosinski et al. 2014]: Initialize from well-trained models improve generalization.
  - **Fine-tuning:** common in deep learning
  - **CV:** ImageNet pre-training.
  - **NLP:** BERT [Devlin et al. 2019], GPT-3 [Brown et al. 2020] …
Outline

• Introduction
• Urban Spatio-temporal Machine Learning
• Transfer Learning
• Urban Spatio-temporal Transfer Learning
  • Spatial Transfer Learning
  • Temporal Transfer Learning
  • Cross-modal Transfer Learning
• Conclusions and Future Work
Urban Spatio-temporal Transfer Learning

• **When to apply, what to transfer** in urban STTL?
  - Planning for new cities.
  - Transfer from existing cities: **spatial STTL**.
  - Planning for new urban services.
  - Transfer from existing urban services: **cross-modal STTL**.
  - Data evolution.
  - Adapt previous knowledge to the present: **temporal STTL**.

Development in Greater Bay Area:

**Spatial STTL**

Constructing a new MTR line:
**Cross-modal STTL**

Adaptation to COVID-19: **Temporal STTL**

[ICAO, 2020]

2021/8/30
Urban Spatio-temporal Transfer Learning

• **Key Challenges:**
  - Follows from urban STML & transfer learning.
  - Common spatio-temporal patterns
  - Common multi-modal knowledge
Spatial STTL

• **Formulation:**
  - Target: city with limited data
  - Source: city with abundant data of the same type
  - Homogeneous TL: \( P(X)_S \neq P(X)_Y \) or \( P(y|X)_S \neq P(y|X)_T \)
    - Differences in urban layout, etc.

• **Coarse VS Fine**-grained methods:
  - **Coarse**-grained methods: Treat each city as a whole and apply transfer learning
  - **Fine**-grained methods: Divide cities as small regions. Find similar region pairs and transfer between them.
Spatial STTL: Coarse-grained

- **Feature-based**: Extract features and project into common space.
  - Commonly multi-modal features: POI, land usage, transportation, etc.

CoFA [Liu et al. 2018a]
- **Problem**: Inferring dockless bike distribution
- **Transfer method**: Factor Analysis,\( \mathcal{H}_{s:t}^*, \mathcal{W}^* , \mu^* = \arg \min_{\mathcal{H}_{s:t}, \mathcal{W}, \mu} \| \mathcal{M}_{s:t} - \mathcal{W}\mathcal{H}_{s:t} - \mu \|_F^2 \),
  - Project original features into a unified feature space by minimizing reconstruction error.

[Pang et al. 2020] and [He et al. 2020]
- **Problem**: Inferring human mobility in new cities
- **Transfer method**: TCA [Pan et al. 2010]
Spatial STTL: Coarse-grained (Cont)

- **Model-based methods**: Train a model on source, re-use on target.
  - Two parts of models: general VS city-specific.

PR-UIDT [Ding et al. 2019]

- **Problem**: Cross-city, cross-user POI recommendation
- **Transfer method**: Regularization
  - Split both user and POI embeddings into general and non-local parts.
  - Regularize on the non-local part.

\[
L_{PR} = \min_{\{P_i^t, Q_i^t\}, \{P_i^n, Q_i^n\}, \{P_n^t, Q_n^t\}} \frac{1}{N} \sum_{i=1}^{N} L_1(P_i^t, Q_i^t) + \alpha L_2(P_i^n, Q_i^n) + L_3(P_n^t, Q_n^t) + \beta \|Q_i^t - Q_i^n\|_F^2.
\]

Matrix Factorization Loss
Regularization on non-local part
Spatial STTL: Fine-grained

• Transfer may lead to higher error than non-transfer:
  • **Negative Transfer:** when domains differ a lot.
  • e.g. DC → NYC.
    DC is far less populated than NYC.

• Solution: Fine-grained methods
  • **Idea:** Cities may be dissimilar, but they must have similar parts (e.g. residential areas, business areas). These parts share common knowledge.
  • **Methodology:** Divide-and-match
    • Divide cities into smaller regions.
    • Obtain similar region pairs.
    • Transfer between similar regions instead of whole cities.

---

[Wang et al. 2019b]

<table>
<thead>
<tr>
<th></th>
<th>D.C.→Chicago 1-day</th>
<th>3-day</th>
<th>Chicago→D.C. 1-day</th>
<th>3-day</th>
<th>D.C.→NYC 1-day</th>
<th>3-day</th>
<th>NYC→D.C. 1-day</th>
<th>3-day</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Data Only</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARIMA</td>
<td>0.740</td>
<td>0.694</td>
<td>0.707</td>
<td>0.661</td>
<td>0.360</td>
<td>0.341</td>
<td>0.707</td>
<td>0.661</td>
</tr>
<tr>
<td>DeepST</td>
<td>0.771</td>
<td>0.711</td>
<td>1.075</td>
<td>0.767</td>
<td>0.350</td>
<td>0.359</td>
<td>1.075</td>
<td>0.767</td>
</tr>
<tr>
<td>ST-ResNet</td>
<td>0.914</td>
<td>0.703</td>
<td>0.869</td>
<td>0.738</td>
<td>0.376</td>
<td>0.349</td>
<td>0.869</td>
<td>0.738</td>
</tr>
<tr>
<td><strong>Source &amp; Target Data</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeepST (FT)</td>
<td>0.652</td>
<td>0.611</td>
<td>0.672</td>
<td>0.619</td>
<td>0.363</td>
<td>0.369</td>
<td>0.713</td>
<td>0.711</td>
</tr>
<tr>
<td>ST-ResNet (FT)</td>
<td>0.667</td>
<td>0.615</td>
<td>0.695</td>
<td>0.623</td>
<td>0.385</td>
<td>0.349</td>
<td>0.696</td>
<td>0.691</td>
</tr>
</tbody>
</table>
Spatial STTL: Fine-grained

- Option 1: Using **fixed** similarity of raw features $M_S, M_t$.
  - Compute region-wise similarity $\rho_{r_t, r_s}$.
  - Minimize domain distance for matched regions $\Delta = \{(r_t, r_s), \forall r_t\}$.

CityTransfer [Guo et al. 2018]
- **Problem**: Cross-city site recommendation
  - Similarity Measure: Pearson correlation
  - Train autoencoder $f$ to minimize
    \[
    \sum_{(r_t, r_s) \in \Delta} \rho_{r_t, r_s} \| f(M_{r_s}) - f(M_{r_t}) \|^2
    \]
    and use $f(M_S), f(M_t)$ for recommendation.

RegionTrans [Wang et al. 2019b]
- **Problem**: Spatio-temporal forecasting
  - Similarity Measure: Cosine similarity
  - Pre-train CNN-LSTM $f_2(f_1(x))$ on source.
  - Fine-tune $f_2(f_1(x))$ on target:
    \[
    \sum_{r_t} \| y_{r_t} - f_2(f_1(x_{r_t})) \|^2 + \sum_{(r_s, r_t) \in \Delta} \rho_{r_s, r_t} \| f_1(x_{r_t}) - f_1(x_{r_s}) \|^2
    \]
    Target label
    Matched domain distance
Spatial STTL: Fine-grained (Cont)

- Option 2: Using **trainable** similarity of output features.

**MetaST [Yao et al. 2019]**
- **Problem**: Spatio-temporal forecasting.
- **Methods**: Transfer **attention** values
  - Cluster source regions using k-means
  - Set memory $M \in \mathbb{R}^{k \times f}$ for each cluster.
  - For source region $r_c$, use output $h_{r_c,k_c}$ to query $M$, get weights $p_{r_c,k_c}$. Match $p_{r_c,k_c}$ with its cluster id.
  - Use attention output $z_{r_c,k_c}$ as complement features for target regions.

**WANT [Liu et al. 2019]**
- **Problem**: Cross-city site recommendation
- **Methods**: Transferability **re-weighting**.
  - Architecture: DANN [Ganin et al. 2016]
  - Weight each source sample with target via:
    - **Domain similarity**: According to $D$.
    - **Data quality**: According to $(G, C)$.
  - Minimize **weighted** DANN loss:

$$\min_G \max_D V(G, C, D) = \frac{1}{n_s} \sum_{i=1}^{n_s} w_{i}^{s}(G, C) - \lambda \left( \frac{1}{n_t} \sum_{i=1}^{n_t} w_{i}^{t}(G, D) + \frac{1}{n_t} \sum_{i=1}^{n_t} l_{d}^{t}(G, D) \right),$$
Spatial STTL: Fine-grained (Cont)

• Summary of fine-grained methods

<table>
<thead>
<tr>
<th>Related Work</th>
<th>Task</th>
<th>Matching Data</th>
<th>Matching Metric</th>
<th>Trainable Matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Wang et al. 2019b]</td>
<td>Spatio-temporal Forecasting</td>
<td>Raw features</td>
<td>Cosine Similarity</td>
<td>No</td>
</tr>
<tr>
<td>[Yao et al. 2019]</td>
<td></td>
<td></td>
<td>Implicitly via Attention</td>
<td>Yes</td>
</tr>
<tr>
<td>[Guo et al. 2018]</td>
<td>Cross-city site recommendation</td>
<td>Raw features</td>
<td>Pearson correlation</td>
<td>No</td>
</tr>
<tr>
<td>[Liu et al. 2021]</td>
<td></td>
<td></td>
<td>Implicitly via domain and label classifier</td>
<td>Yes</td>
</tr>
</tbody>
</table>

• Raw feature matching:
  • **Pro**: Stable. Incorporates multi-modal information.
  • **Con**: Performance relies on quality of features.

• Output feature matching:
  • **Pro**: Flexible, trainable matching.
  • **Con**: Limited target data \(\rightarrow\) overfit.

[Check-in (auxiliary) is related to crowd flow (limited).]
Temporal STTL

- Temporal STTL tackles data **distribution shifts**.
  - Two kinds of distribution shifts

- **Formulation**:  
  - Homogeneous TL  
  - Source: Previous periods  
  - Target: Current period

- **Related Concept: Continual** learning [Chen and Liu, 2018].  
  - A series of domains and tasks $\mathcal{D}_i$, $\mathcal{T}_i$, $i = 1 \ldots N$,  
  - For each $n$, learn task $\mathcal{T}_n$ with knowledge from $\mathcal{T}_i$, $i < n$.  
  - In temporal STTL, each timestamp defines a task.

[Sudden changes, e.g. holiday [Zhang et al. 2017]

[Long-term (monthly) changes [Karacasu et al. 2011]
Temporal STTL for Indoor Localization

- Indoor localization: signal strength $s = (s_1, \ldots, s_m) \rightarrow$ location $(x, y)$
- Temporal TL for indoor localization:
  - **Goal**: At $t_k$, adapt $f_1$ to $f_k$ using $\{s_{ki}, l_{ki}\}_{i=1}^l, \{s_{kj}\}_{j=l+1}^{l+u}$
  - The first $l$ labeled data are from fixed positions: **Reference points**.

- **Feature-based** method: LeManCoR [Pan et al. 2007]
  - Localization with extended **Manifold Co-regularization**
  - **Assumption**: Models from different times **agree on reference points**.

$$f_1^*, f_k^* = \arg\min_{f_1 \in H_{K_1}, f_k \in H_{K_k}} \frac{\mu}{l} \sum_{i=1}^l V(s_{1i}, l_{1i}, f_1) + \gamma_A^{(1)} \| f_1 \|_{H_{K_1}} + \gamma_B^{(1)} \| f_1 \|_I + \frac{1}{l} \sum_{i=1}^l V(s_{ki}, l_{ki}, f_k) + \gamma_A^{(2)} \| f_k \|_{H_{K_k}} + \gamma_B^{(2)} \| f_k \|_I + \frac{\gamma_I}{l} \sum_{i=1}^l [f_1(s_{1i}) - f_k(s_{ki})]^2,$$

- $|f|_H$: Complexity regularization
- $|f|_I$: Manifold regularization
Temporal STTL for Indoor Localization (Cont)

• **Model-based** method: TrHMM [Zheng et al. 2008]
  • Transferring Hidden Markov Models: \((L, O, \lambda, A, \pi)\)
    • \(L, O\): Location & observation space
    • \(\lambda = P(o|l) = N(\mu, \Sigma), l \in L, o \in O\)
    • \(A\): Transition matrix between locations \(L\). \(\pi\): Initial distribution over \(L\)
• **Assumption**: Fixed relations between reference (r) and other points (k) across \(t\).
• Method:
  1. Estimate \(\lambda_1\) at time \(t_1\).
  2. Obtain relations between reference and other points via regression.

\[
s_j^k = \alpha_{0j}^k + \alpha_{1j}^k r_{1j} + \ldots \alpha_{ij}^k r_{lj} + \varepsilon_j.
\]
  3. At time \(t\), **re-use** the regression model to reconstruct data, and **fine-tune** \(\lambda_t\)

\[
\mu_t = \beta \mu_1 + (1 - \beta) \mu_t^{reg},
\]

\[
\Sigma_t = \beta (\Sigma_1 + (\mu_t - \mu_1)(\mu_t - \mu_1)^T) + (1 - \beta) (\Sigma_t^{reg} + (\mu_t - \mu_t^{reg})(\mu_t - \mu_t^{reg})^T).
\]
Cross-modal STTL

- **Two scenarios** of cross-modal STTL:
  - Missing feature modality:
    - e.g. Predict air quality using road maps, POI, weather, and transport data, transport data is missing in a city.
    - **Solution**: Learn the relation between existing modalities and the missing one.
  - Heterogeneous TL: $\mathcal{X}_S \neq \mathcal{X}_T$

- Missing label modality:
  - e.g. Detect ride-sharing car trajectories with no labeled trajectories.
  - **Solution**: Find related data (e.g. taxis), and link source labels with target ones.
  - Heterogeneous TL: $\mathcal{Y}_S \neq \mathcal{Y}_T$
Cross-modal STTL: Missing feature modality

• FLORAL [Wei et al. 2016a]
  • Assumption: Relations between modalities are invariant across domains.
  • Key Idea: Extract and transfer such relations
  • Methods:
    • Relation encoding: Build sample-modality graph
      Nodes: Each sample in each modality
      Intra-modality edges: feature distance
      Inter-modality edges: sample proximity
    • Relation learning: Cluster sample-modality graph.
      Each cluster forms base vectors in dictionaries.
    • Transfer: Share dictionaries across domains
      Sparse coding to obtain domain-invariant features.
      Multi-modal TrAdaBoost to reweight modalities.
Cross-modal STTL: Missing label modality

• CoHTL [Wei et al. 2016b]
  • **Target**: sensor data with few labels. **Source**: Social messages
  • **Key Idea**: Link samples & labels from social messages to sensors.
  • **Methods**:
    • **Label linking**: Topic model & word embedding.
    • **Sample linking**: Spatio-temporal proximity.
    • **Feature-based transfer**: Link regularization.

\[
\mathcal{O} = \|P - UV_1\|^2_F + \|Q - WV_2\|^2_F \\
+ \beta \sum_{i=1}^{m} \sum_{j=1}^{n} S_{ij} \|u_i - w_j\|^2_2 + \gamma R(U, W, V_1, V_2),
\]

Regularization by sample links \(S_{ij}\).

Example of spatio-temporal sample linking

Results of label linking by word embeddings.
Summary of urban STTL

- **Spatial**: Homogeneous TL
  - **Coarse VS Fine**-grained methods: transfer between similar parts

- **Temporal**: Homogeneous TL,
  - **Example**: Temporal STTL for indoor localization (via reference points)

- **Cross-modal**: Heterogeneous TL
  - **Cases**: Missing feature/label modality
  - **Key**: Relations between modalities.

<table>
<thead>
<tr>
<th>Urban STTL Setting</th>
<th>Existing Works</th>
<th>Common ST Patterns</th>
<th>Common Multi-modal Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Coarse-grained Feature Model</td>
<td>Fine-grained Instance Feature Model</td>
</tr>
<tr>
<td>Liu et al. [63]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Liu et al. [64]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Pang et al. [65]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>He et al. [66]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Li et al. [67]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ding et al. [68]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Wang et al. [69]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Wang et al. [70]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Yao et al. [71]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Song et al. [72]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mallick et al. [73]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Guo et al. [74]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Liu et al. [75]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Wang et al. [72]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- **Spatial Transfer Learning**

<table>
<thead>
<tr>
<th>Temporal Transfer Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pan et al. [55]</td>
</tr>
<tr>
<td>Zheng et al. [76]</td>
</tr>
</tbody>
</table>

- **Cross-modal Transfer Learning**

<table>
<thead>
<tr>
<th>Wei et al. [77]</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wang et al. [78]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Wei et al. [79]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Conclusion

• **Motivation**: Urban computing + machine learning meets lack of data.
• **Challenge**: Common ST patterns & Common multi-modal knowledge
• **Categorization**: Spatial, temporal and cross-modal
• **Future directions**:
  • Transfer learning with **effective multi-modal fusion**:
    • Existing works mainly use feature concat or feature similarity.
  • Transfer learning with **dynamics**:
    • Temporal STTL with detection of data shifts.
    • Adaptive knowledge transfer at different periods.
  • Transfer learning with **privacy**
    • Spatio-temporal data may contain user privacy.
References


References


References


References


References


