Deep Learning for Human Mobility Analytics

-- L2: Tutorial on DL and Spatio-Temporal Data Sensing

Yuxuan Liang (梁宇轩)

INTR & DSA Thrust

yuxuanliang@hkust-gz.edu.cn

Changes on the Syllabus

	Date	Lecture/Tutorial Topics	Pre	Remarks
1	2 Sep	L1: Overview and Introduction		
2	9 Sep	L2: Tutorials on Deep Learning	✓	Introduce foundations in deep learning techniques, such as CNN, RNN, GNN, etc.
13	16 Sep	Transfer to **an online course for project discussion** (each team 20 mins) in **early November**. Details will be informed in late October.		Mid-Autumn Festival
3	23 Sep	L3: Spatio-Temporal Data Sensing and Management	✓	To form team (2 people)
4	30 Sep	L4: Cross-Domain Data Fusion and Multimodal Learning	✓	
5	14 Oct	L5: Location Embedding and Urban Region Profiling	✓	**Meeting 1**: Submit 1-page plan for project (by **14 Oct @ 23:59**)
6	21 Oct	L6: Learning Spatio-Temporal Trajectory Data	✓	
7	28 Oct	L7: Learning Spatio-Temporal Raster Data	✓	Submit Design Report (by **28 Oct @ 23:59**)
8	4 Nov	L8: Learning Spatio-Temporal Graph Data	✓	
9	11 Nov	L9: Advanced Topics in Human Mobility Analytics I	✓	Cover SSL and transfer learning **Meeting 2**: Finalize designs & applications
10	18 Nov	L10: Advanced Topics in Human Mobility Analytics II	✓	Cover PINNs, reinforcement learning, and adversarial training
11	25 Nov	L11: Large Language Models for Human Mobility Analytics	✓	
12	2 Dec	L12: Summary and Future Trends	✓	* Project presentation * Submit Systems Report & Codes (by **9 Dec @ 23:59**)

Presentation Issues

A	B	C	D	E	F	G	H	I	J	K	
	Topic	**Order**	**Paper Title**	**Paper URL**	**Venue**	**Year**	**Institution (the first one)**	**Presenter**	**Reason to present (e.g., award-level, impact, significance, innovation, potential, etc.)**	**Confirmed by Yuxuan?**	**Que b**
1	**Overview and Introduction**	1	U-Air: When Urban Air Quality Inference Meets Big Data **[This is a Sample Row!!!]**	Link	KDD	2013	MSRA	Yuxuan LIANG	Test-of-time Award at KDD China; Open a new problem to the community.	yes	
2	**Tutorials on Deep Learning**	1									
		2									
3	**Spatio-Temporal Data Sesning and Management**	1	Transformer-based Map Matching Model with Limited Ground-Truth Data using Transfer-L	Link	TRC	2022	McGill University	Jiahui LIANG	This work proposes a method based on transfer learning to solve matching task by limited number of ground-truth data which can be applied on the IoT system.	yes	
		2	TraSS: Efficient Trajectory Similarity Search Based on Key-Value DataStores	Link	ICDE	2022	JD	Gangyong ZHU	This work proposes TraS5S, an efficient framework for trajectory similarity search in key-value data stores.	yes	
		3									
4	**Cross-domain Data Fusion and Multimodal Learning**	1	GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction	Link	IJCAI	2018	Xidian&JD	Qiongyan WANG	This work incorporates the attention mechanism to model the dynamic ST dependencies.	no	
		2	TIME-LLM: TIME SERIES FORECASTING BY REPROGRAMMING LARGE LANGUAGE	Link	ICLR	2024	Monash	Pei Liu	This work introduce the TIME-LLM framework, a framework for reprogramming large language models (LLMs) for general time series forecasting while keeping the base language model intact.	no	
		3	Attention Bottlenecks for Multimodal Fusion	Link	Neurips	2021	Google	Yongzi Yu	This paper proposes an impressive way to realize the fusion between audion and vision, based on the transformer structure.	no	
5	**Location Embeddings and Urban Region Profiling**	1	Beyond the First Law of Geography: Learning Representations of Satellite Imagery by Leveraging Point-of-Interests	https://dl.acm.org/doi/10	WWW '22	2022	University of Helsinki	Yixuan Wang	This paper introduces an innovative framework that integrates POI data with satellite imagery to improve the prediction of socioeconomic indicators, significantly outperforming existing methods.	yes	
		2	Koopa: Learning Non-stationary Time Series Dynamics with Koopman Predictors	Link	NIPS	2023	Qinghua	Jiaxi Hu	In this paper, koopman theory is introduced, which treats time series data as the observations of the same dynamic system, and uses grid embedding to get the representation, and then learns the linear evolution operator in the high-dimensional Koopman space.	no	
		3	Urban Region Profiling With Spatio-Temporal Graph Neural Network	.org/stamp/stamp.jsp?tp	IEEE	2022	Dalian University of Technology	Tianyu Wei	This paper is selected for presentation due to its innovative approach in utilizing spatio-temporal graph neural networks for urban region profiling, which is a significant advancement in the field of urban analytics and has potential applications in smart city development.	yes	
6	**Learning Spatio-Temporal Trajectory Data**	1		Link	CIKM	2022	NUS	Jingtao HE	This work employs transformer structure which is widely used in NLP field into trajectory classification task. To achieve this goal, the adaptive improvement have been made for the characteristics of the trajectory classification task.		
		2	Planning with Diffusion for Flexible Behavior Synthesis	Link	ICML	2022	Berkeley	Ruiguo ZHONG	This work show how classifier-guided sampling and image inpainting can be reinterpreted as coherent planning strategies.	no	
		3	Modeling Trajectories with Recurrent Neural Networks	Link	IJCAI	2017	Singapore Management University	Zhixiong Wang	This paper is worth presenting because it introduces an innovative use of Recurrent Neural Networks (RNNs) to model trajectory data, enhancing prediction accuracy and having broad applications in fields like urban mobility and autonomous driving.	yes	

Selected Projects @ Previous Semesters

- Trajectory data mining
  - Exploring trajectories' representation capability in graph structure
  - Estimated Time of Arrival Prediction based on LaDe
  - Predicting Urban EV Charging Demand Using Trajectory Data
  - Personalized Vehicle Energy Consumption Prediction
  - Exploring Interpretable Dynamic Graph Neural Networks for Human Mobility Forecasting
- Planning and policy making
  - Planning for electrical vehicles charging stations: A data-driven approach using spatio-temporal data with consideration of multiple factors
  - Predicting optimal charging station locations based on traffic volume
  - Optimal gate control strategies for metro system: A community-detection-integrated ADMM-based approach
  - Optimization of Delivery Routing and Site Selection for Express Logistics

Selected Projects

- Spatio-temporal forecasting
  - Predicting electric vehicle charging demand using spatio-temporal data mining and analytics
  - Knowledge Distillation for Spatio-Temporal Graph Neural Networks
  - Spatio-Temporal Graph Forecasting Boosted by Large Language Models
  - Urban Traffic Prediction from Spatio-Temporal Data Using LLMs
  - Conformal Predictions on Spatio-Temporal Graphs
  - Prediction of Highway Traffic Flow: A Decomposition-based Spatio-Temporal Attention Fusion Network
  - Spatio-Temporal Mixture-of-Experts for Traffic Prediction
- Location embedding and region profiling
  - Multimodal Pretraining in Region Embedding
- Applications
  - Improving DS-MSA in AirFormer for Higher Air Quality Prediction Accuracy
  - Traffic inductive learning for blockout missing

Self-Introduction

- To let us know each student, please feel free to introduce yourself (2-3 minutes), covering especially
  - Your name and thrust
  - Academic background
  - Research interest
  - You (potential) advisor
  - What project do you plan to do in this course?
  - Some other interesting experience to share…

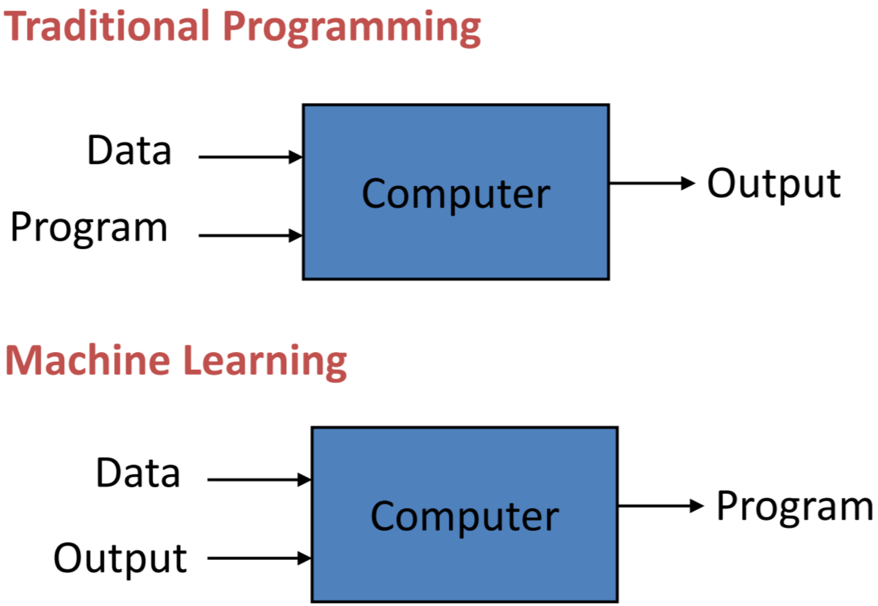
Preliminary

To introduce

- Machine learning foundations
- Deep learning foundations

What is Machine Learning?

- “Learning is any process by which a system improves performance from experience.”

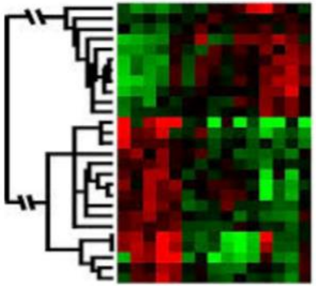


https://www.seas.upenn.edu/~cis5190/fall2017/lectures/01_introduction.pdf

When Do We Use Machine Learning?

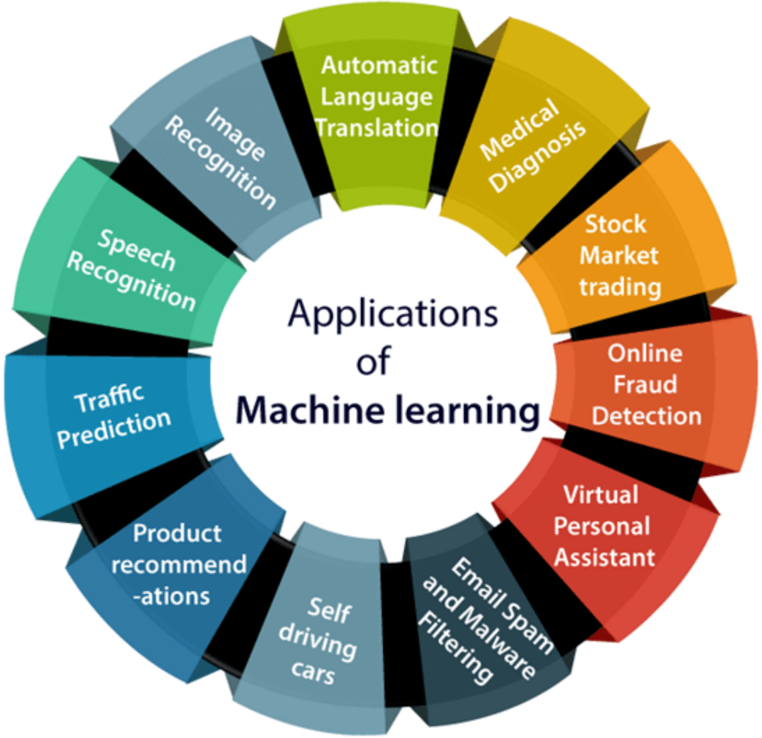
- ML is used when:
  - Human expertise does not exist (navigating on Mars)
  - Human cannot explain their expertise (speech recognition)
  - Models must be customized (personalized medicine)
  - Models are based on huge amount of data (genomics)



Sample Applications

- Web search
- Robotics
- Social networks
- Urban computing
- E-commerce
- Recommendation system
- …

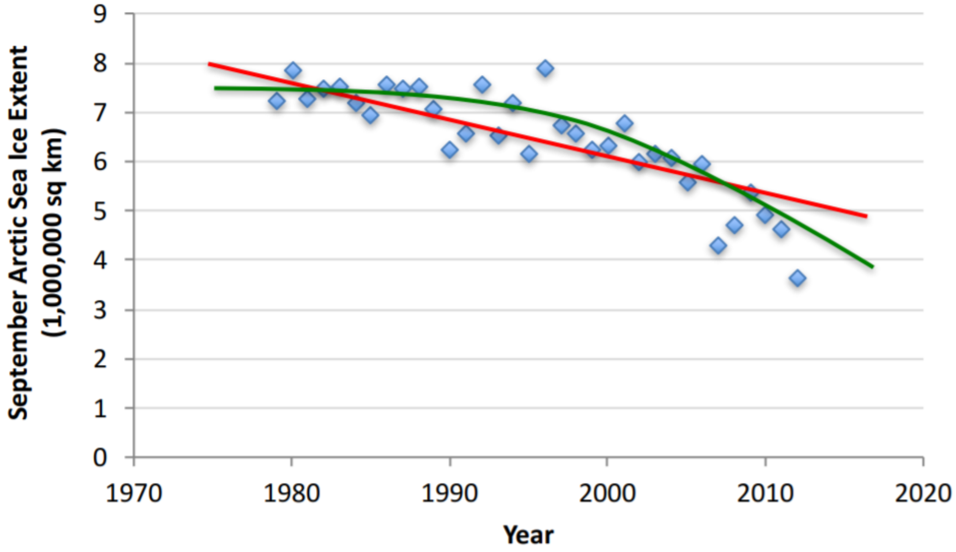


Types of Learning

- Supervised learning
  - Given: training data + desired outputs (labels)
- Unsupervised learning
  - Given: training data (without desired outputs)
- Semi-supervised learning
  - Given: training data + a few desired outputs
- Reinforcement learning
  - Rewards from sequence of actions

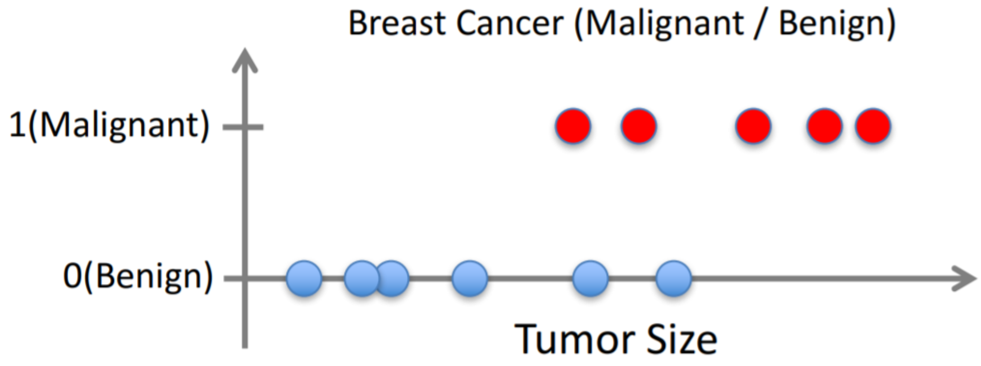
Supervised Learning: Regression

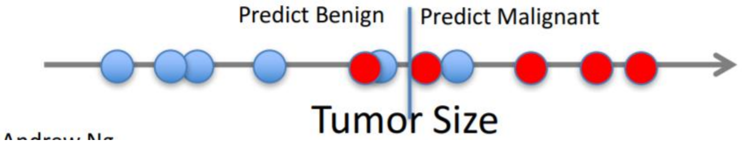
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is real-valued == regression



Supervised Learning: Classification

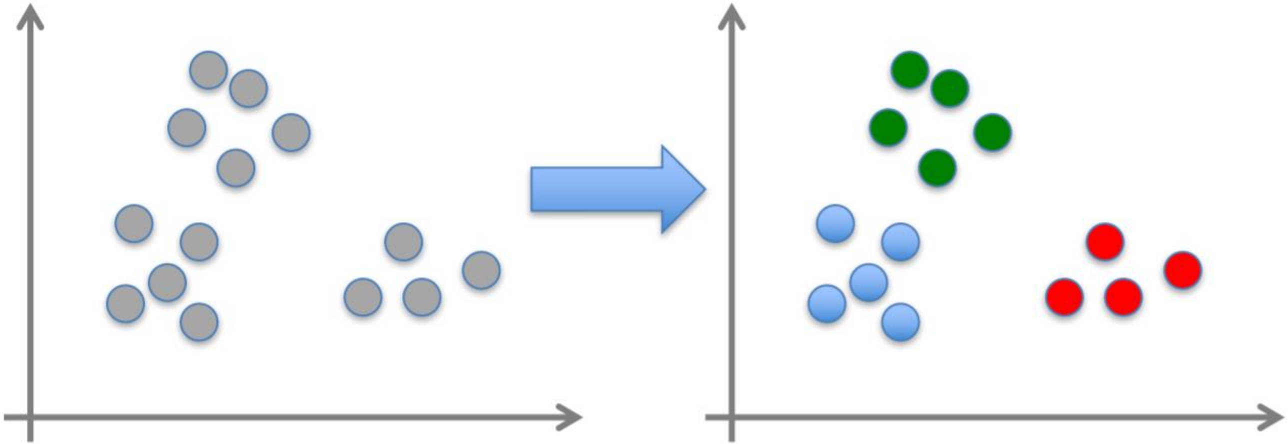
- Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification





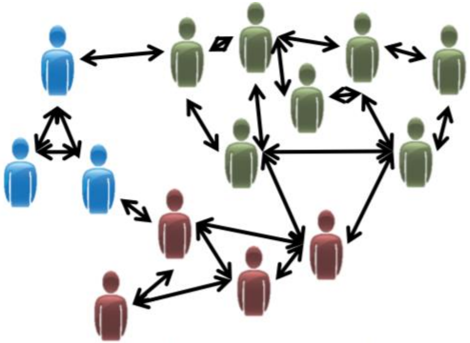
Unsupervised Learning

- Given $x_1, x_2, \dots, x_n$ (without labels)
- Output hidden structure behind the x's
  - E.g., clustering



Unsupervised Learning

Organize computing clusters

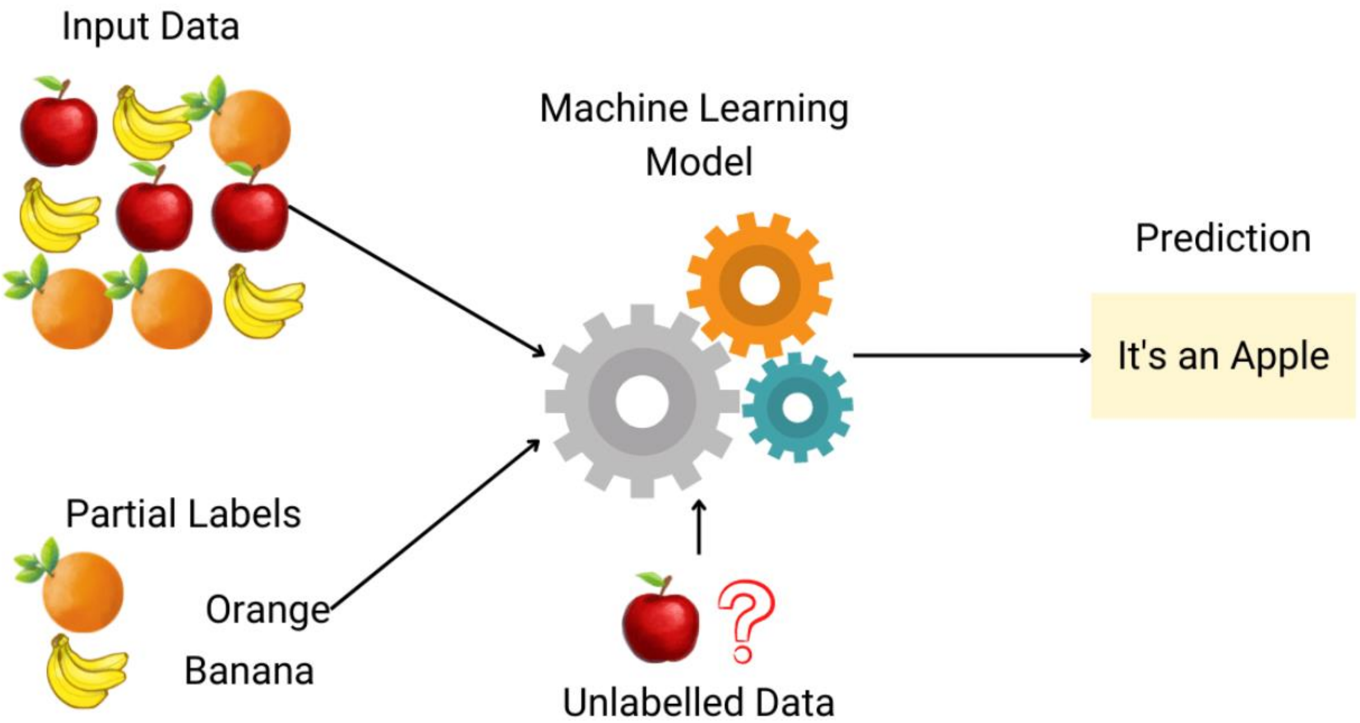


Social network analysis

Market segmentation

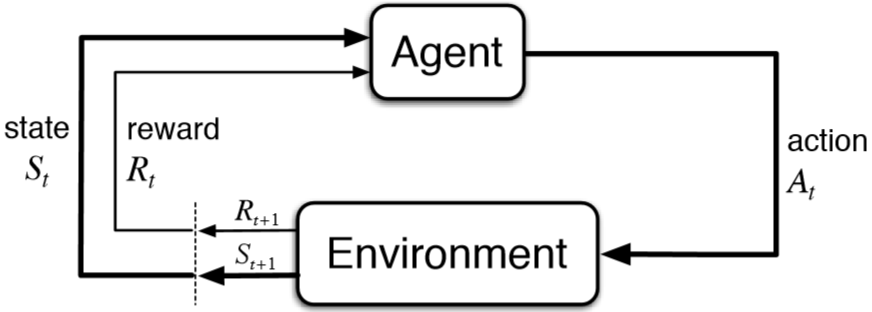
Astronomical data analysis

Semi-Supervised Learning



Reinforcement Learning

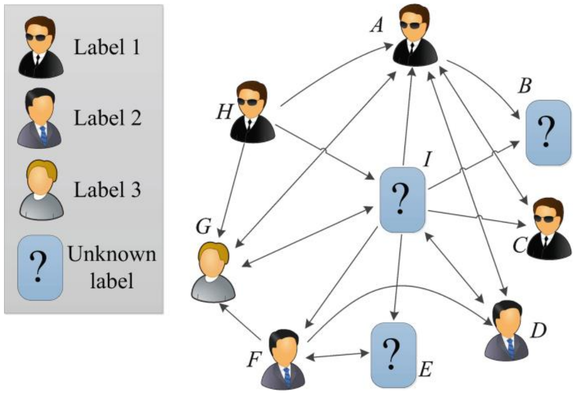
- Given a sequence of states and actions with (delayed) rewards, output a policy
  - Policy is a mapping from states -> actions that tells you what to do in a given state
- Examples:
  - Credit assignment problem
  - Game playing
  - Robot in a maze
  - Balance a pole on your hand

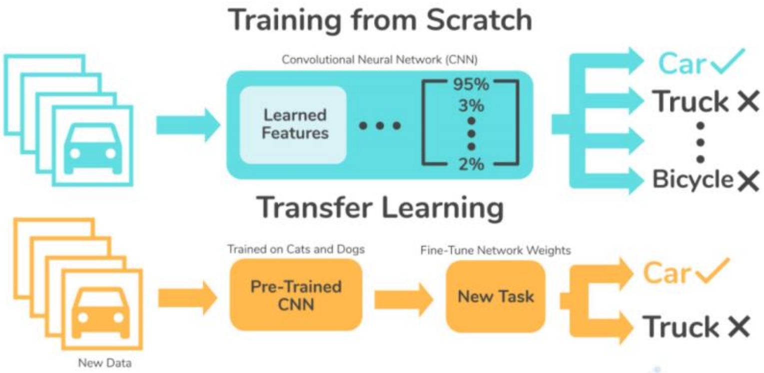


Reinforcement Learning Applications

Training vs Test Distribution

- We generally assume that the training and test examples are independently drawn from the same overall distribution of data
  – We call this “i.i.d” which stands for “independent and identically distributed”
- If examples are not independent, requires collective classification
- If test distribution is different, requires transfer learning





ML in a Nutshell

- Every ML algorithm has three components:
  - Representation
  - Optimization
  - Evaluation

Various Function Representations

- Numerical functions
  - Linear regression
  - Neural networks
  - Support vector machines
- Symbolic functions
  - Decision trees
  - Rules in propositional logic
  - Rules in first-order predicate logic
- Instance-based functions
  - Nearest-neighbor
  - Case-based
- Probabilistic Graphical Models
  - Naïve Bayes
  - Bayesian networks
  - Hidden-Markov Models (HMMs)
  - Probabilistic Context Free Grammars (PCFGs)
  - Markov networks

Various Optimization Methods

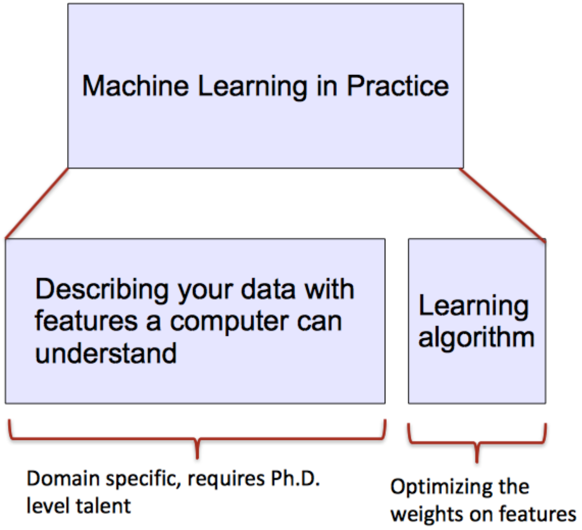
- Gradient descent
  - Perceptron
  - Backpropagation
- Dynamic Programming
  - HMM Learning
  - PCFG Learning
- Divide and Conquer
  - Decision tree induction
  - Rule learning
- Evolutionary Computation
  - Genetic Algorithms (GAs)
  - Genetic Programming (GP)
  - Neuro-evolution

Evaluation Metrics

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Cost / Utility
- Margin
- Entropy
- KL-divergence

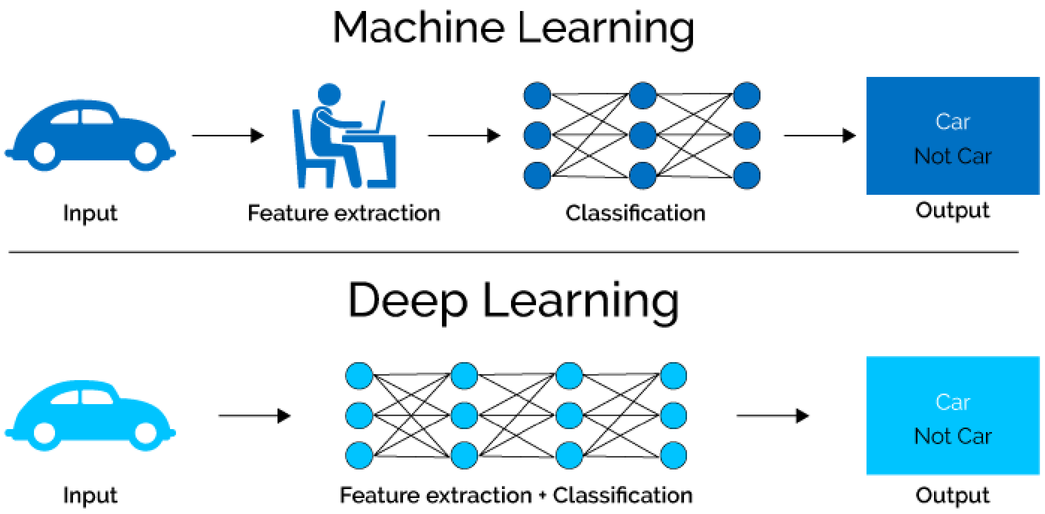
ML vs. Deep Learning

- Most machine learning methods work well because of human-designed representations and input features
- ML becomes just optimizing weights to best make a final prediction



What is Deep Learning (DL)?

- A machine learning subfield of learning representations of data. Exceptional effective at learning patterns
- Deep learning algorithms attempt to learn (multiple levels of) representation by using a hierarchy of multiple layers

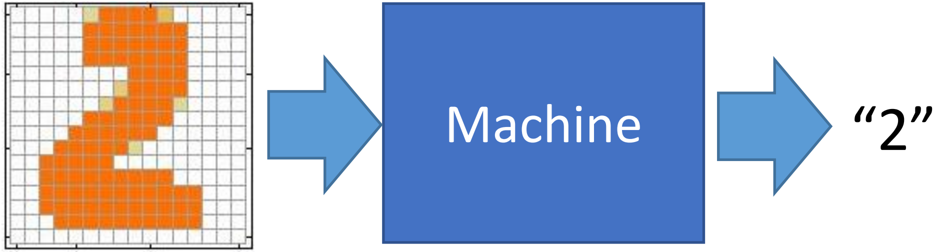


Why is DL useful?

- Manually designed features are often over-specified, incomplete and take a long time to design and validate
- Learned Features are easy to adapt, fast to learn
- Deep learning provides a very flexible, (almost?) universal, learnable framework for representing world, visual and linguistic information.
- Can learn both unsupervised and supervised
- Effective end-to-end joint system learning
- Utilize large amounts of training data

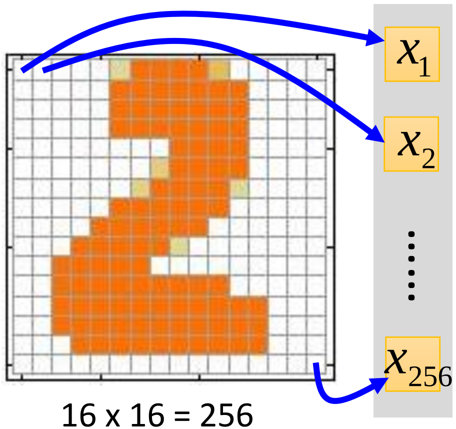
Example Application

- Handwriting Digit Recognition



Handwriting Digit Recognition

Input

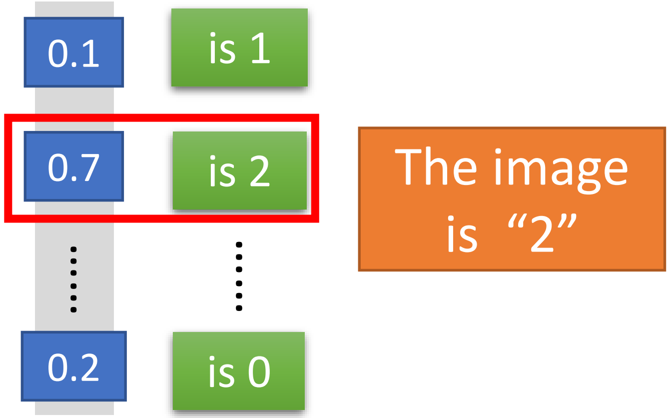


16 x 16 = 256

Ink → 1

No ink → 0

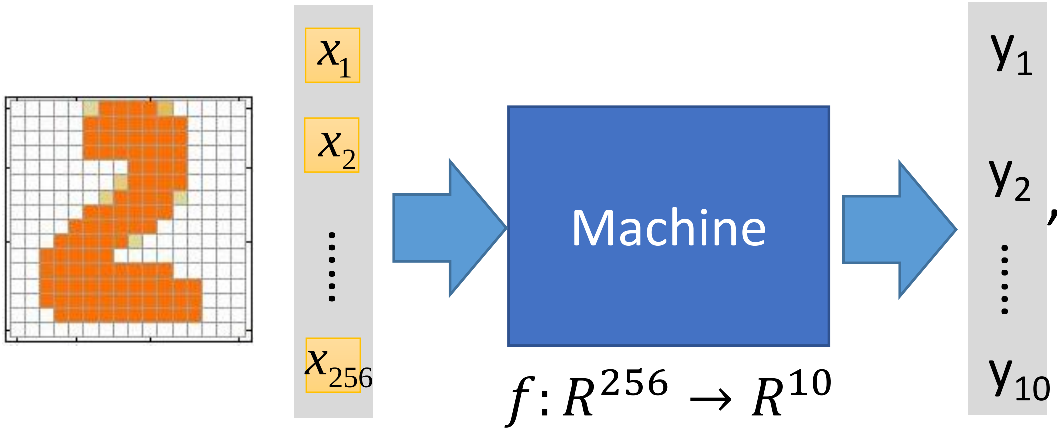
Output



Each dimension represents the confidence of a digit.

Example Application

- Handwriting Digit Recognition

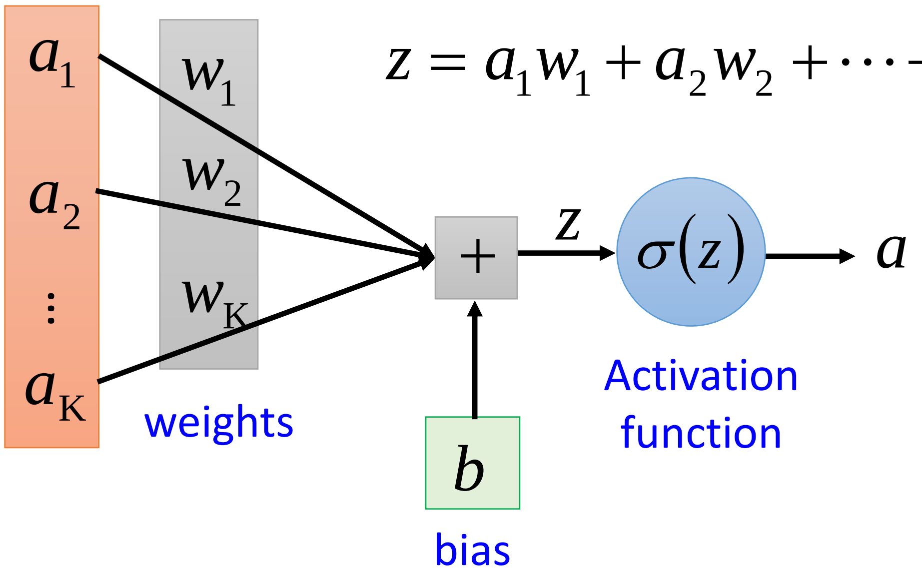


In deep learning, the function f is represented by neural network

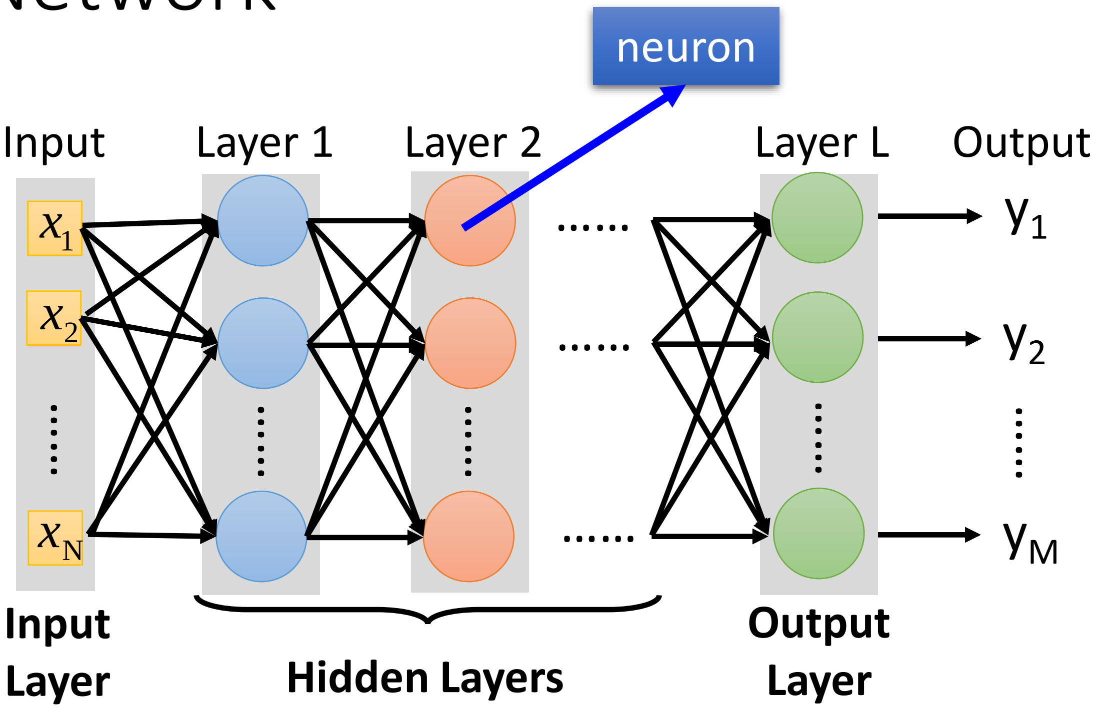
Element of Neural Network

<u>Neuron</u> $f: R^K \rightarrow R$

$$z = a_1 w_1 + a_2 w_2 + \cdots + a_K w_K + b$$

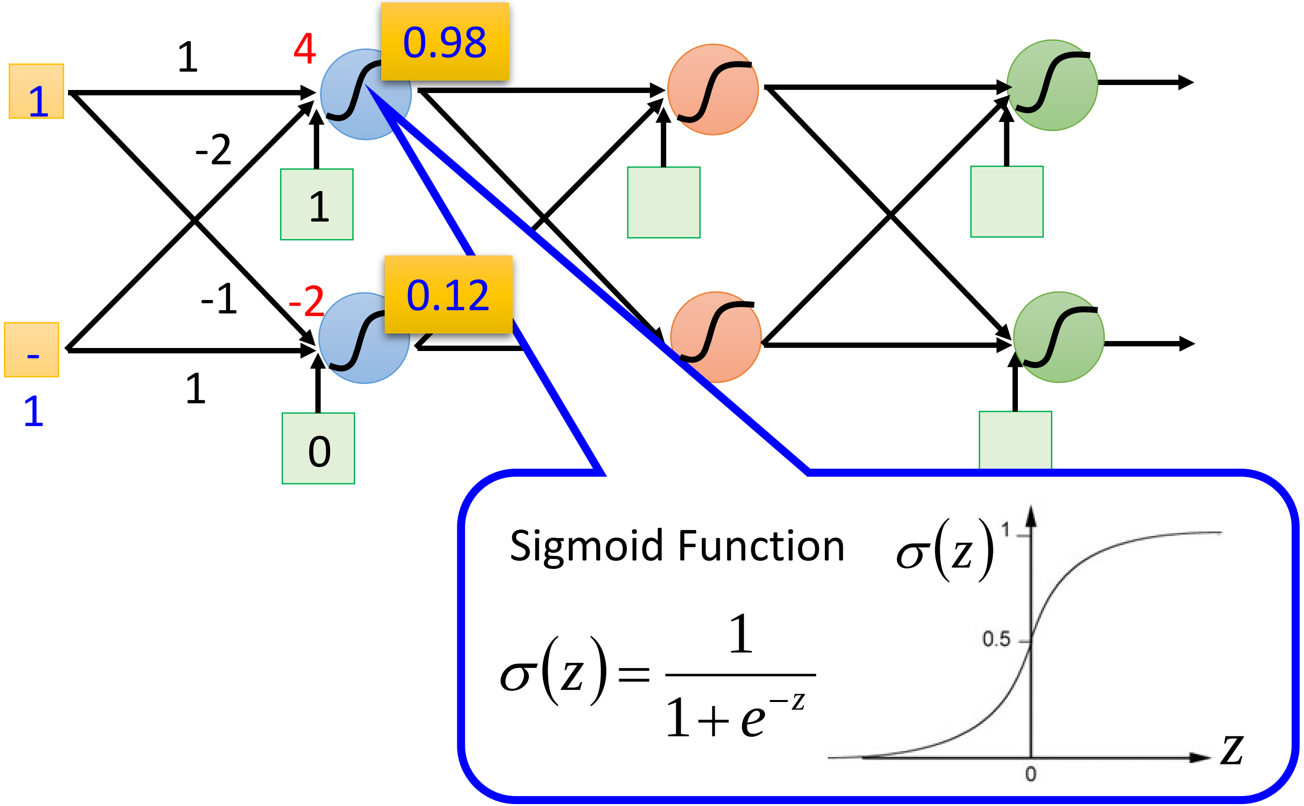


Neural Network



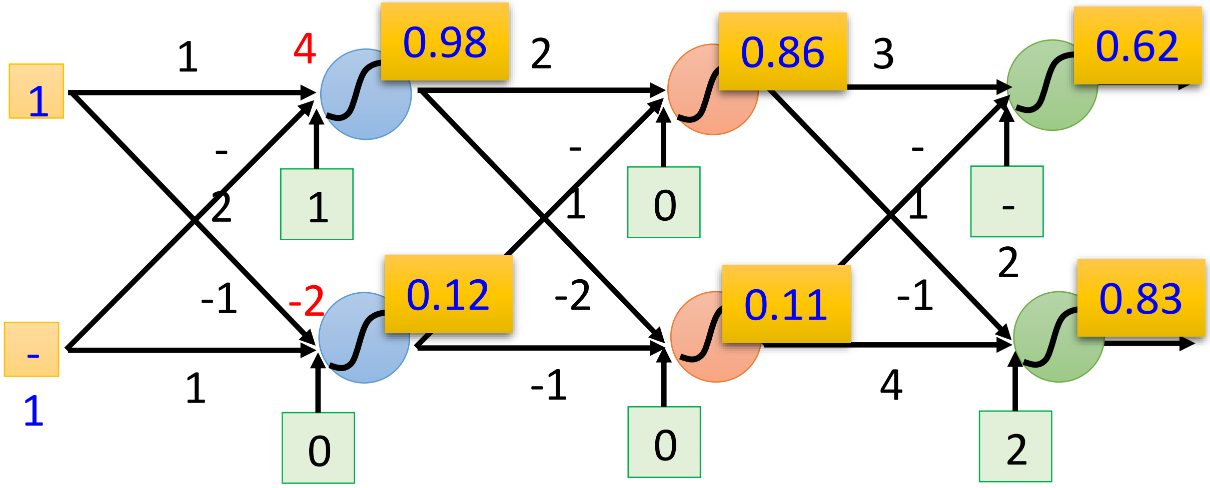
Deep means many hidden layers

Example of Neural Network

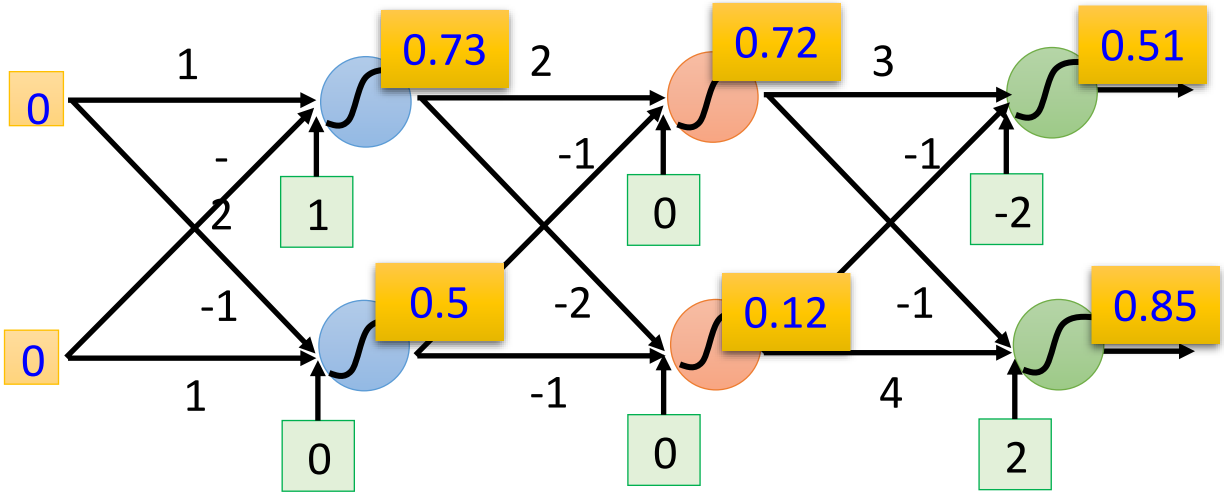


$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Example of Neural Network



Example of Neural Network

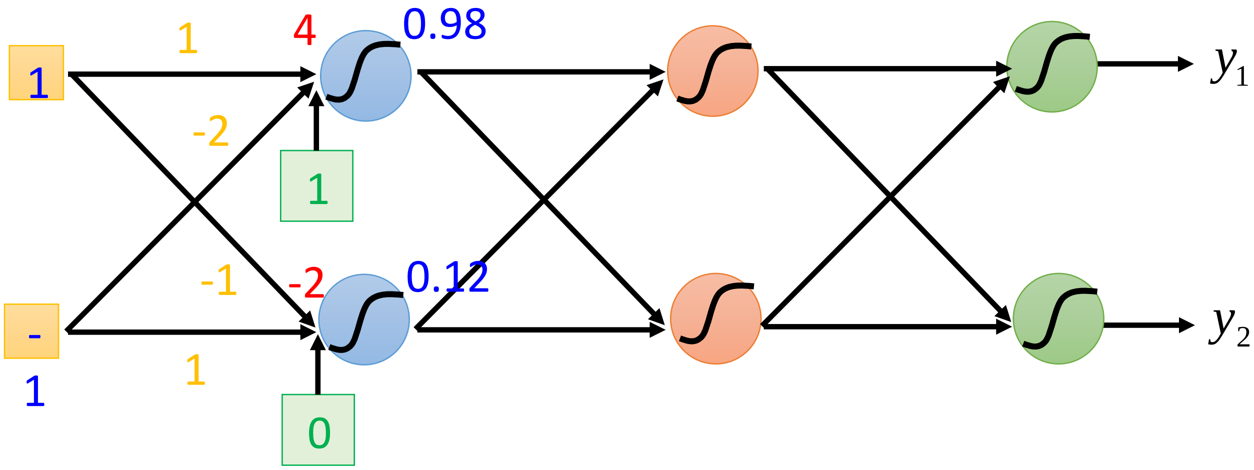


$f: R^2 \to R^2$

$$f\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) = \begin{bmatrix} 0.62 \\ 0.83 \end{bmatrix} \quad f\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0.51 \\ 0.85 \end{bmatrix}$$

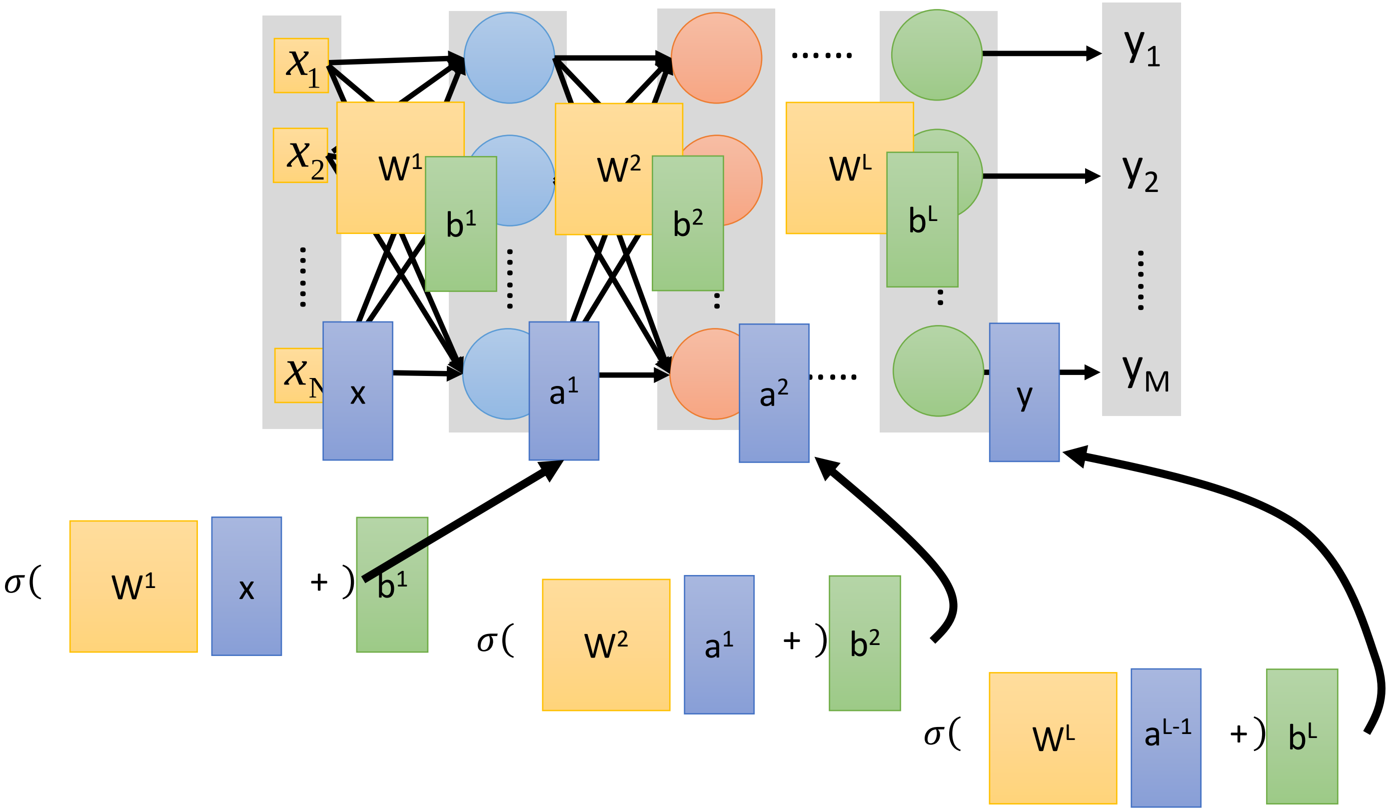
Different parameters define different function

Matrix Operation

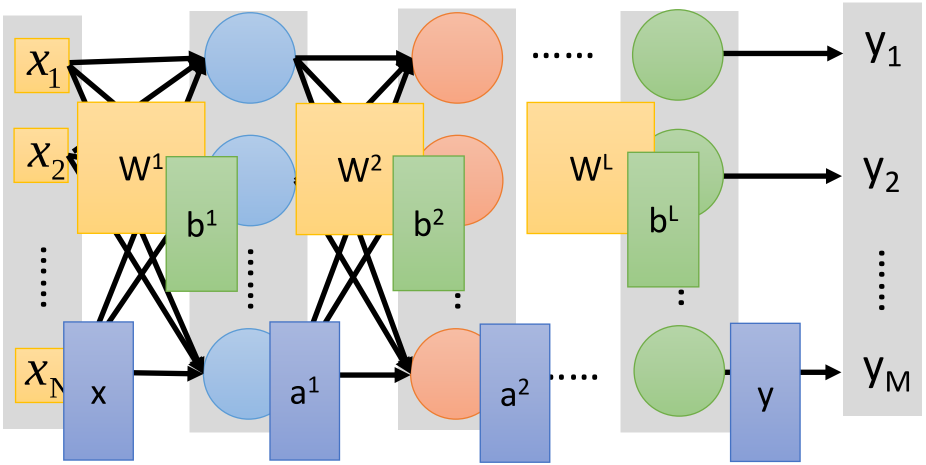


$$\sigma\left(\underbrace{\begin{bmatrix}1 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}+\begin{bmatrix}1\\ 0\end{bmatrix}}_{\begin{bmatrix}4\\ -2\end{bmatrix}}\right)=\begin{bmatrix}0.98\\ 0.12\end{bmatrix}$$

Neural Network



Neural Network



$$y = f(x)$$

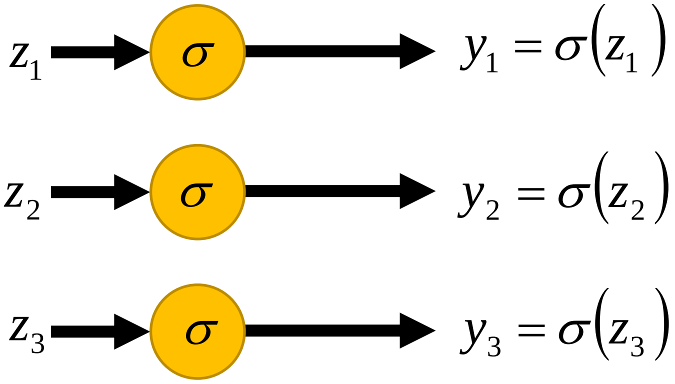
Using parallel computing techniques to speed up matrix operation

$$= \sigma(W^L \cdots \sigma(W^2 \sigma(W^1 x + b^1) + b^2) \cdots + b^L)$$

Softmax

- Softmax layer as the output layer

<u>Ordinary Layer</u>



In general, the output of network can be any value.

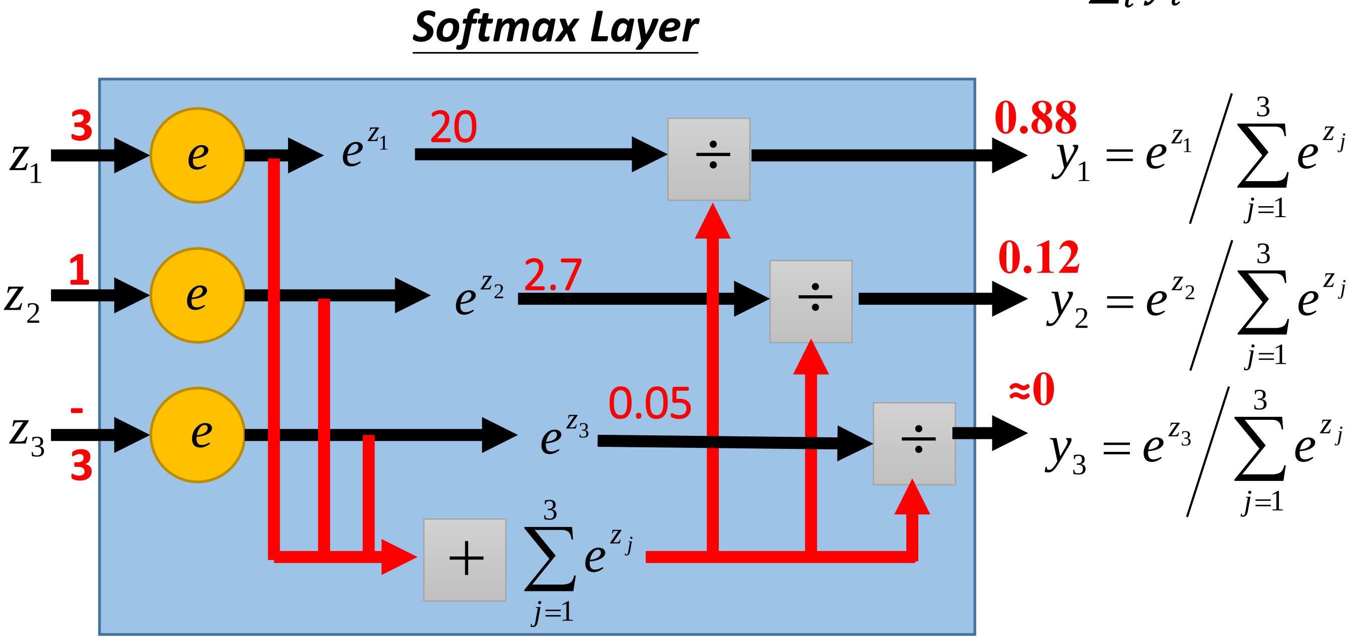
May not be easy to interpret

Softmax

- Softmax layer as the output layer

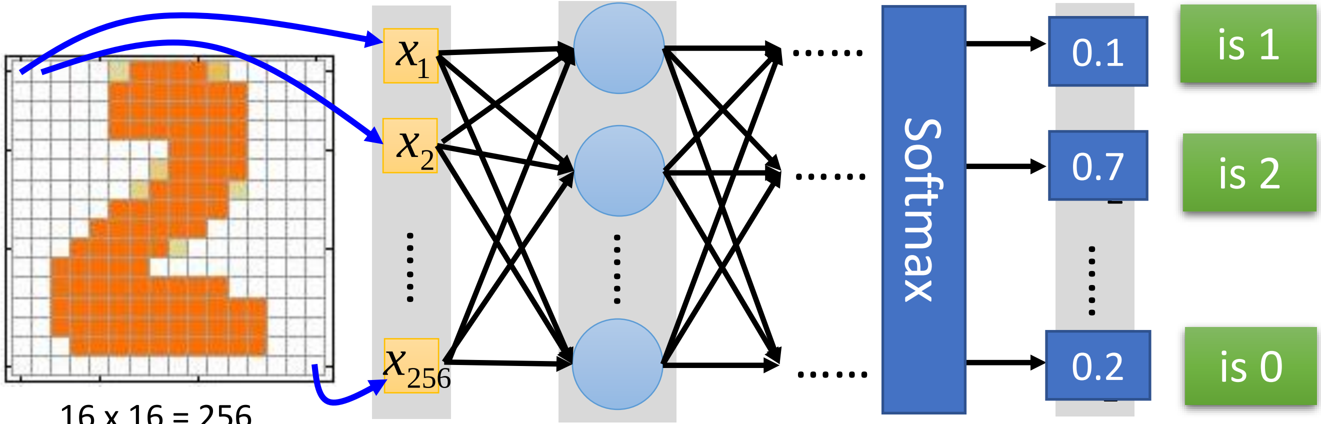
Probability:

- $1 > y_i > 0$
- $\sum_i y_i = 1$



How to set network parameters

$$\theta = \{W^1, b^1, W^2, b^2, \cdots W^L, b^L\}$$



Ink → 1

No ink → 0

Set the network parameters θ such that ……

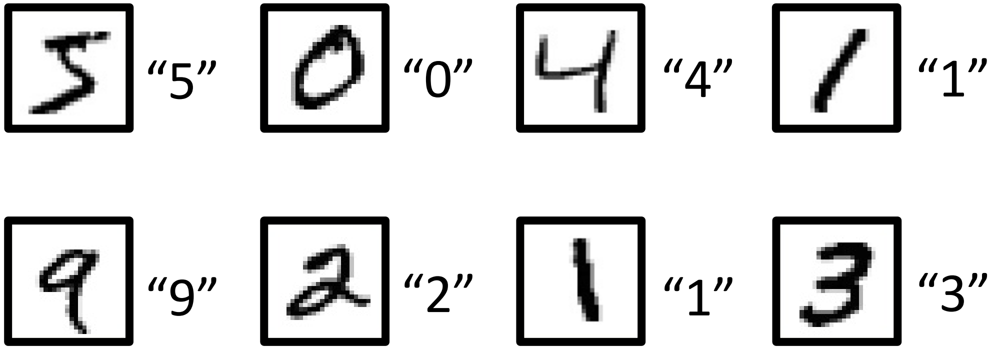
Inpu ... m value

How to let the neural network achieve this

Input:  y_2 has the maximum value

Training Data

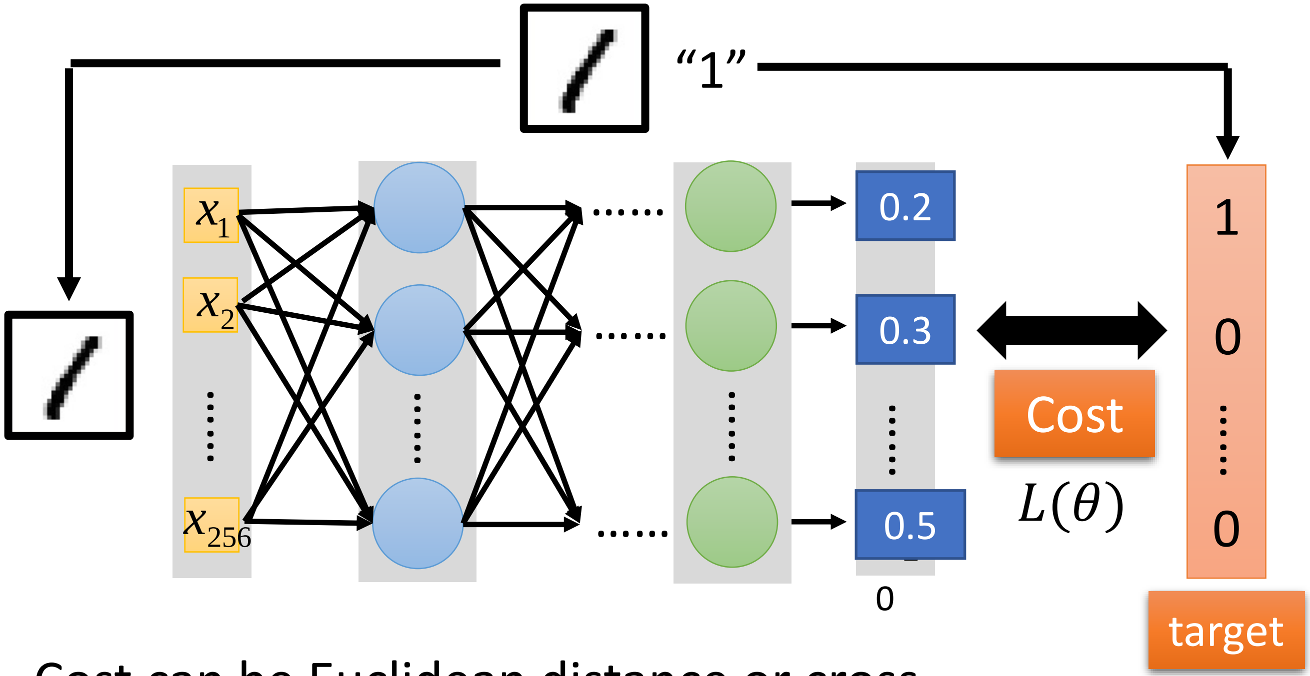
- Preparing training data: images and their labels



Using the training data to find the network parameters.

Cost

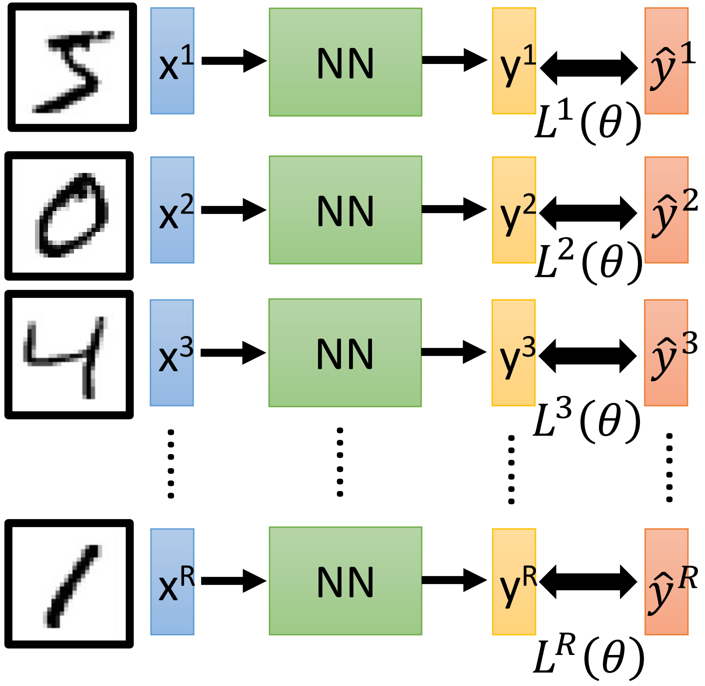
Given a set of network parameters θ, each example has a cost value.



Cost can be Euclidean distance or cross entropy of the network output and target

Total Cost

For all training data …



Total Cost:

$$C(\theta) = \sum_{r=1}^{R} L^r(\theta)$$

How bad the network parameters θ is on this task

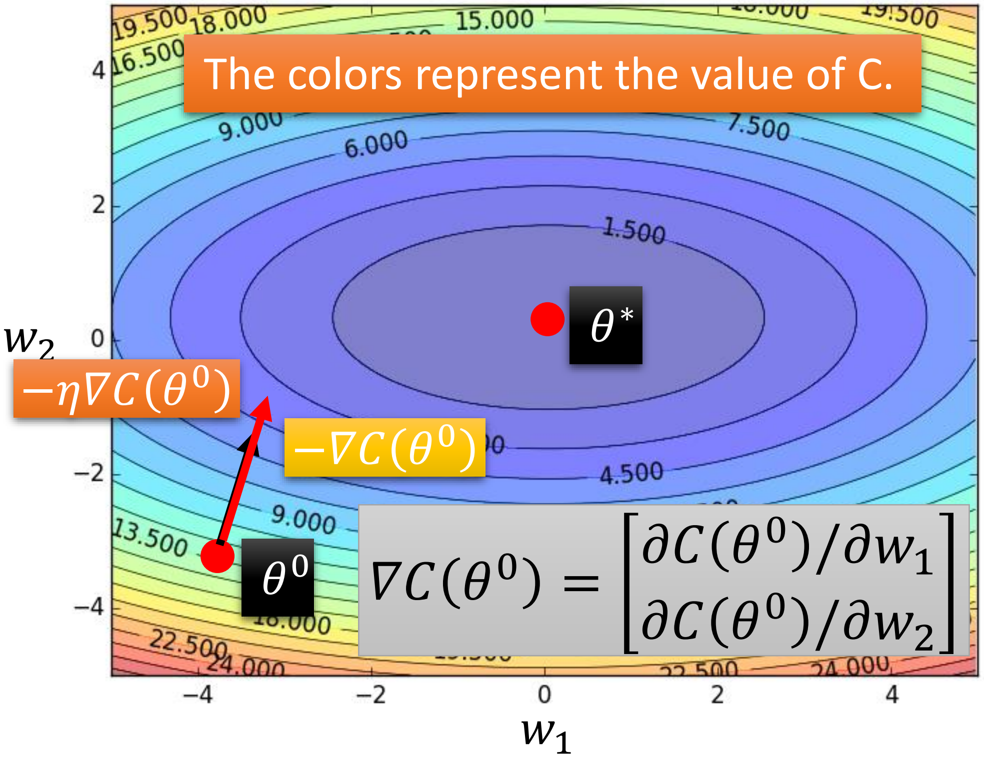
Find the network parameters θ^* that minimize this value

Gradient Descent

Assume there are only two parameters w_1 and w_2 in a network.

$$\theta = \{w_1, w_2\}$$

Error Surface



Randomly pick a starting point θ^0

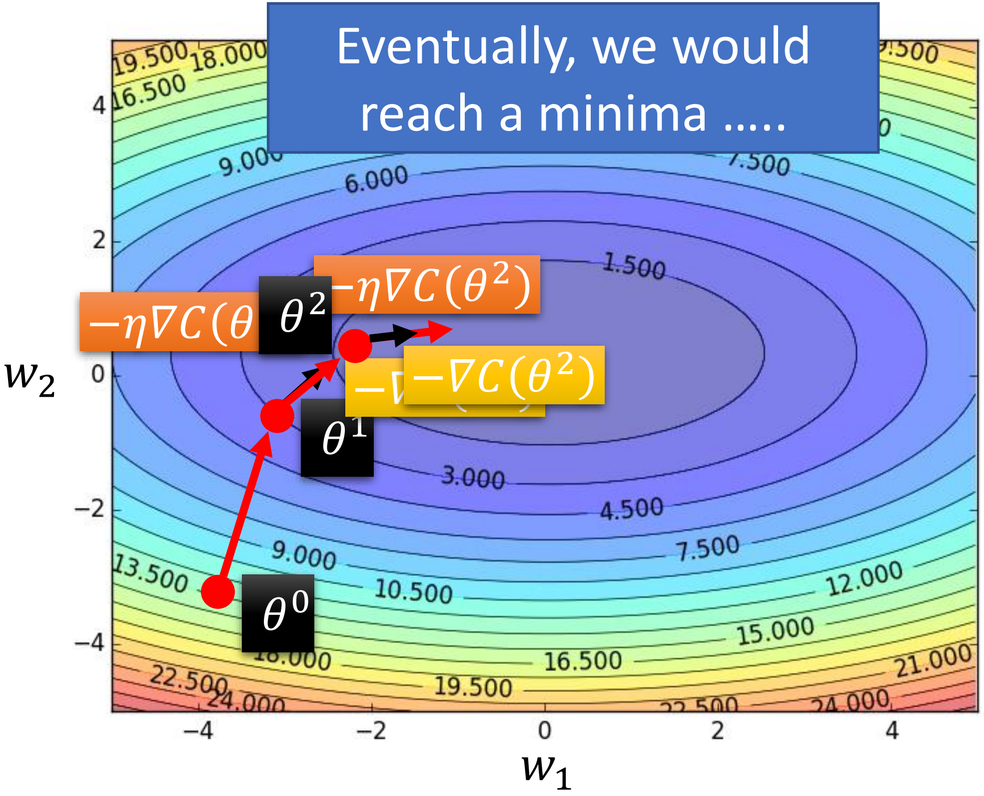
Compute the negative gradient at θ^0

$\Rightarrow -\nabla C(\theta^0)$

Times the learning rate η

$\Rightarrow -\eta\nabla C(\theta^0)$

Gradient Descent



Randomly pick a starting point θ^0

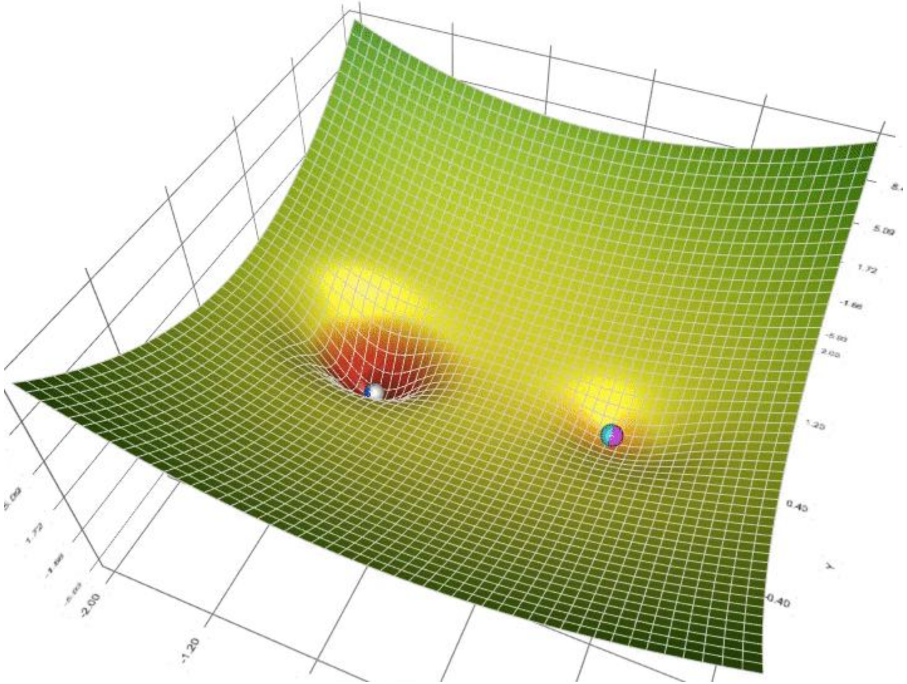
Compute the negative gradient at θ^0

$\Rightarrow -\nabla C(\theta^0)$

Times the learning rate η

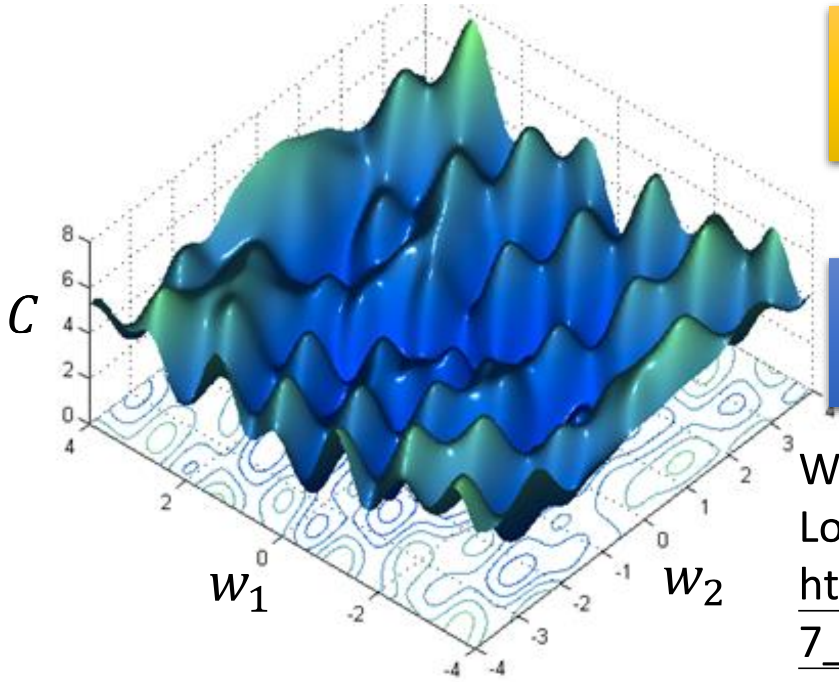
$\Rightarrow -\eta\nabla C(\theta^0)$

Graident Descent



Local Minima

- Gradient descent never guarantee global minima

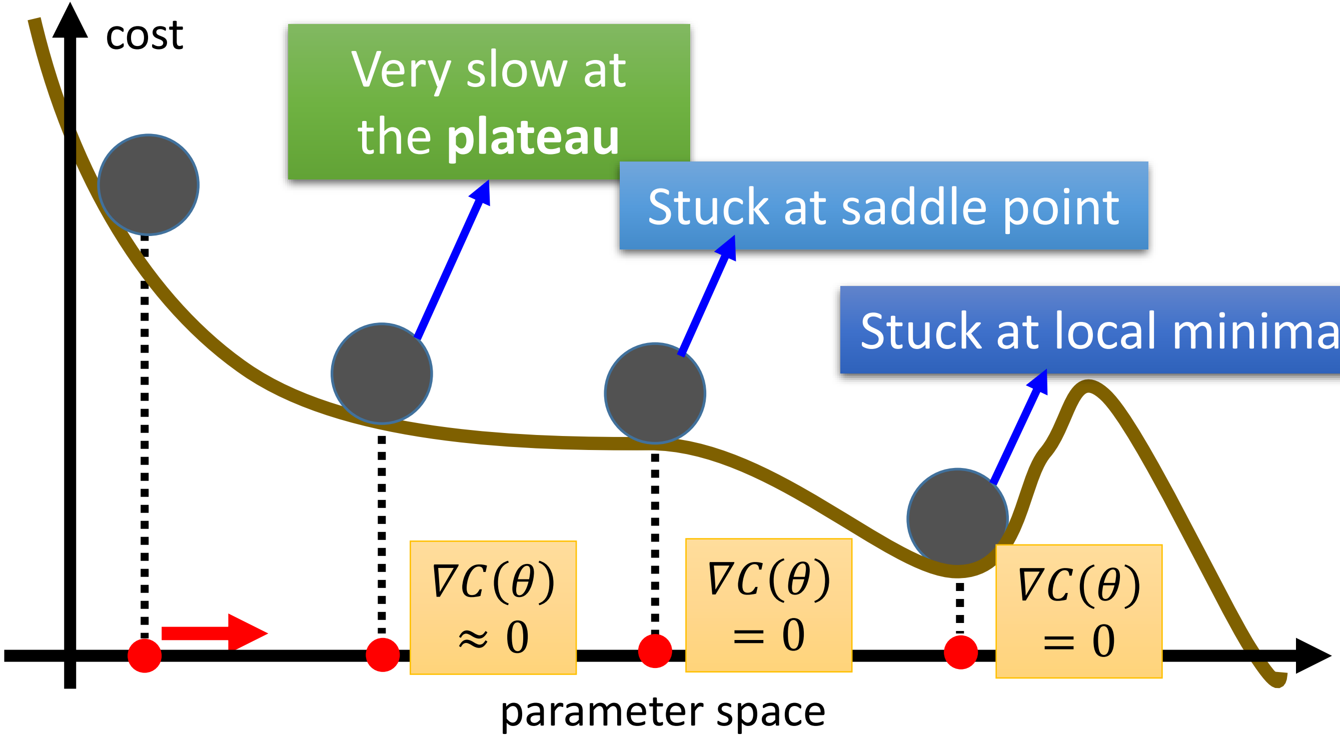


Different initial point θ^0

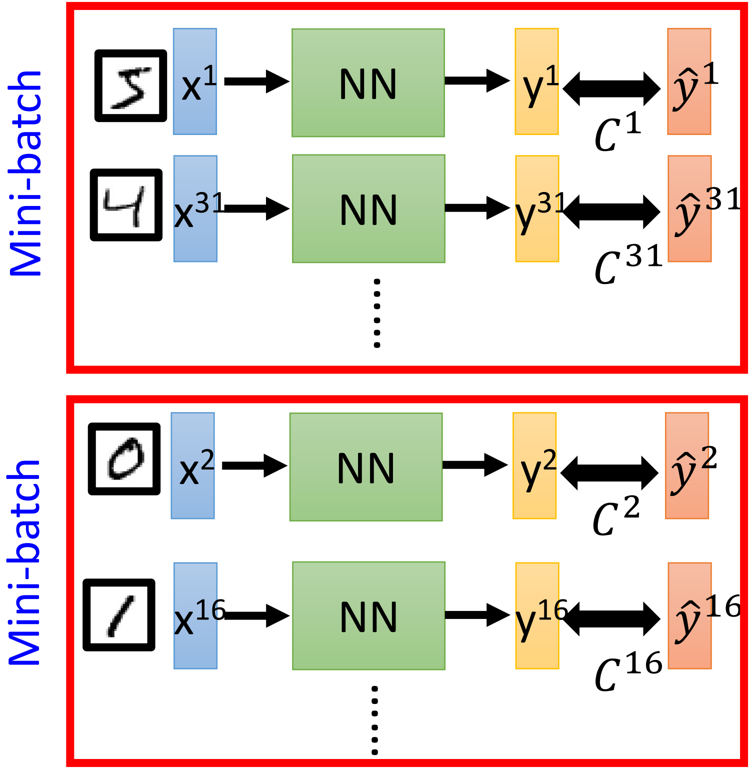
Reach different minima, so different results

Who is Afraid of Non-Convex Loss Functions?
http://videolectures.net/eml07_lecun_wia/

Besides local minima ……



Mini-batch



- Randomly initialize θ^0
- Pick the 1st batch
  $C = C^1 + C^{31} + \cdots$
  $\theta^1 \leftarrow \theta^0 - \eta \nabla C(\theta^0)$
- Pick the 2nd batch
  $C = C^2 + C^{16} + \cdots$
  $\theta^2 \leftarrow \theta^1 - \eta \nabla C(\theta^1)$
  ⋮
- Until all mini-batches have been picked

one epoch

Repeat the above process

Backpropagation

- A network can have millions of parameters.
    - Backpropagation is the way to compute the gradients efficiently (not today)
- Many toolkits can compute the gradients automatically

Size of Training Data

- Rule of thumb:
  - the number of training examples should be at least five to ten times the number of weights of the network.

- Other rule:

$$N > \frac{|W|}{(1-a)}$$

$|W|$= number of weights

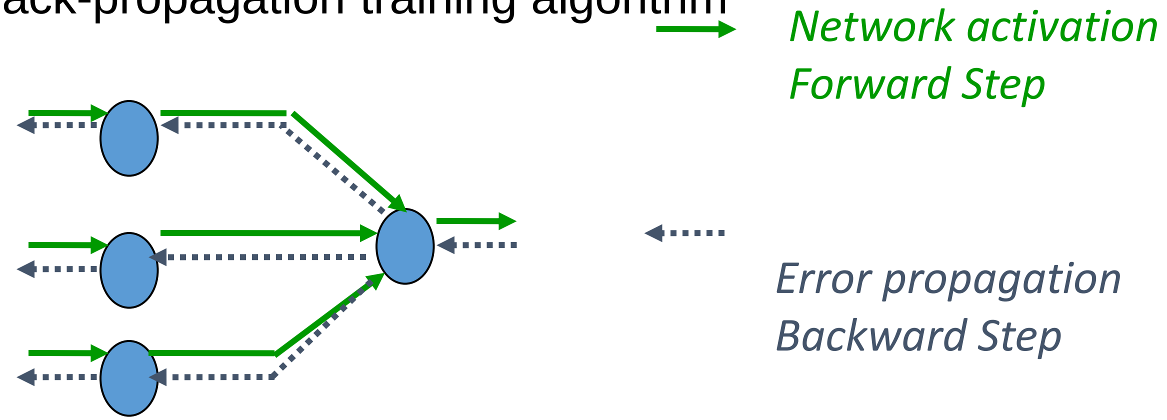
a = expected accuracy on test set

Training: Backprop algorithm

- The Backprop algorithm searches for weight values that minimize the total error of the network over the set of training examples (training set).
- Backprop consists of the repeated application of the following two passes:
  - **Forward pass**: in this step the network is activated on one example and the error of (each neuron of) the output layer is computed.
  - **Backward pass**: in this step the network error is used for updating the weights. Starting at the output layer, the error is propagated backwards through the network, layer by layer. This is done by recursively computing the local gradient of each neuron.

Back Propagation

- Back-propagation training algorithm



- Backprop adjusts the weights of the NN in order to minimize the network total mean squared error.

Why Deep?

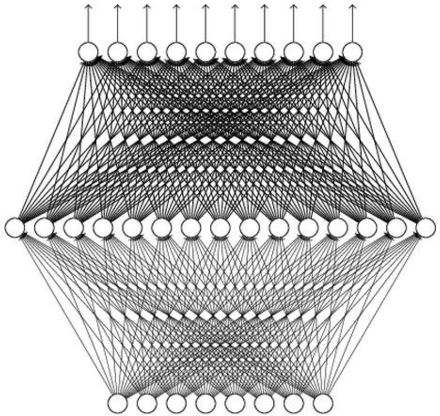
Universality Theorem

Any continuous function f

$$f : R^N \rightarrow R^M$$

Can be realized by a network with one hidden layer

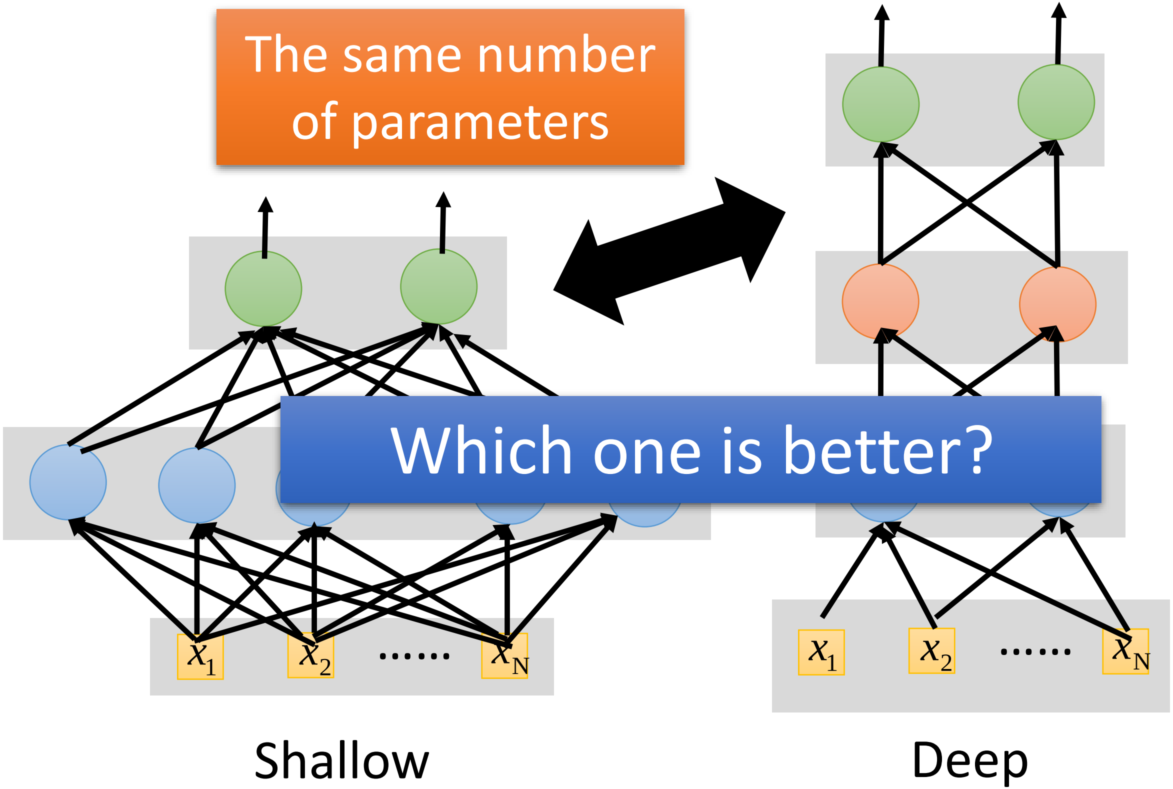
(given **enough** hidden neurons)



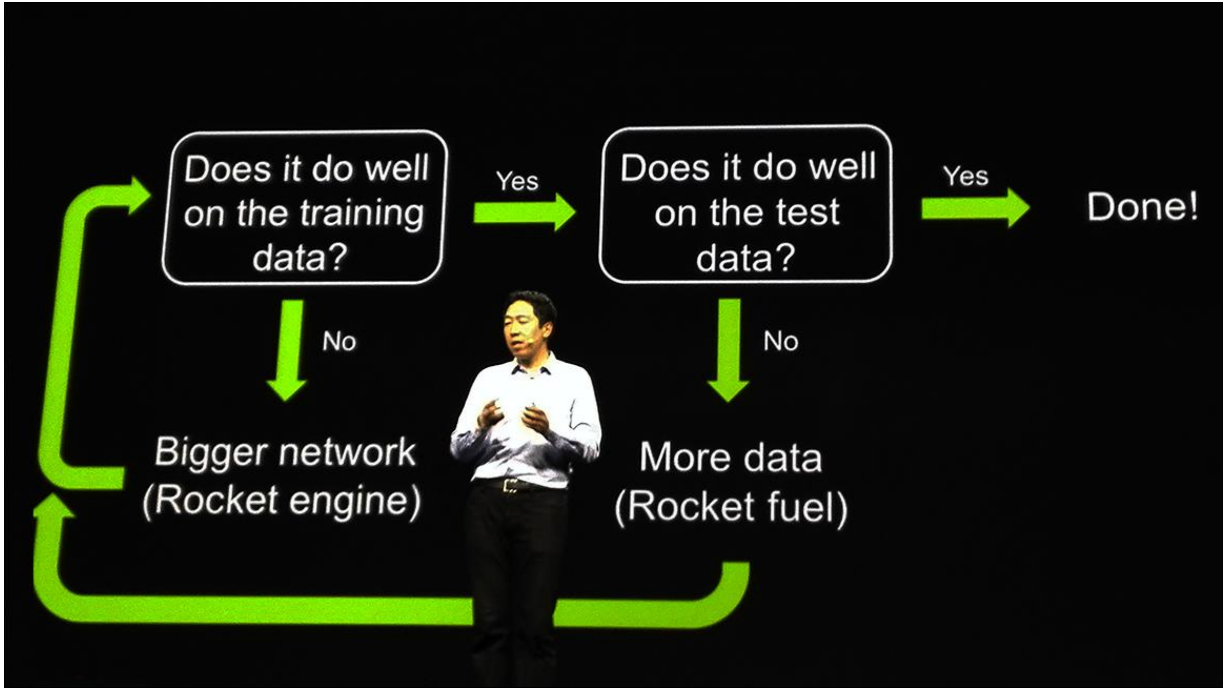
Reference for the reason: http://neuralnetworksanddeeplearning.com/chap4.html

Why "Deep" neural network not "Fat" neural network?

Fat + Short v.s. Thin + Tall

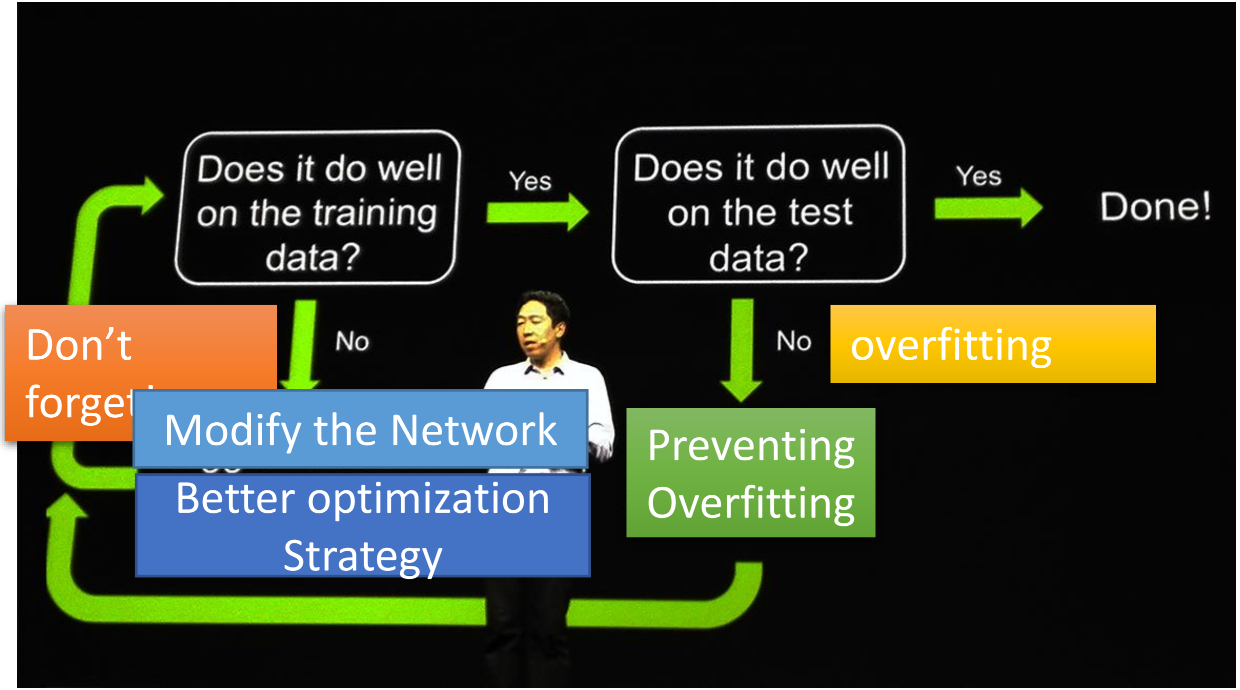


Recipe for Learning



http://www.gizmodo.com.au/2015/04/the-basic-recipe-for-machine-learning-explained-in-a-single-powerpoint-slide/

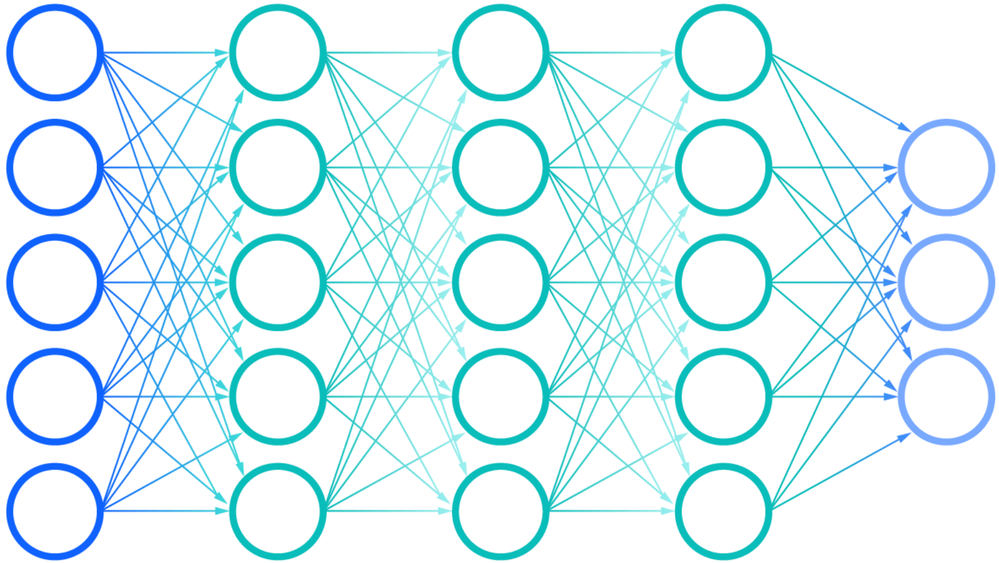
Recipe for Learning



http://www.gizmodo.com.au/2015/04/the-basic-recipe-for-machine-learning-explained-in-a-single-powerpoint-slide/

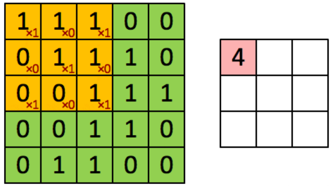
Multi-Layer Perceptron (MLP)

- Input layers
- Hidden layers
- Output layers



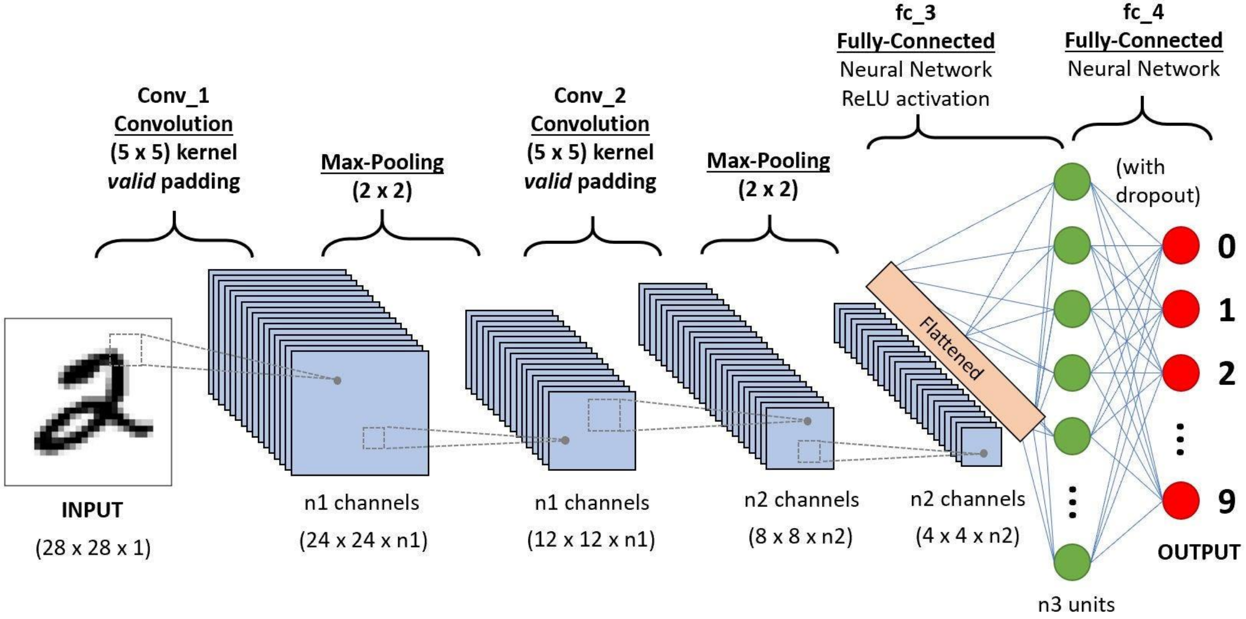
Convolutional Neural Networks (CNN)

- Convolution
- Max-pooling
- FC/MLP

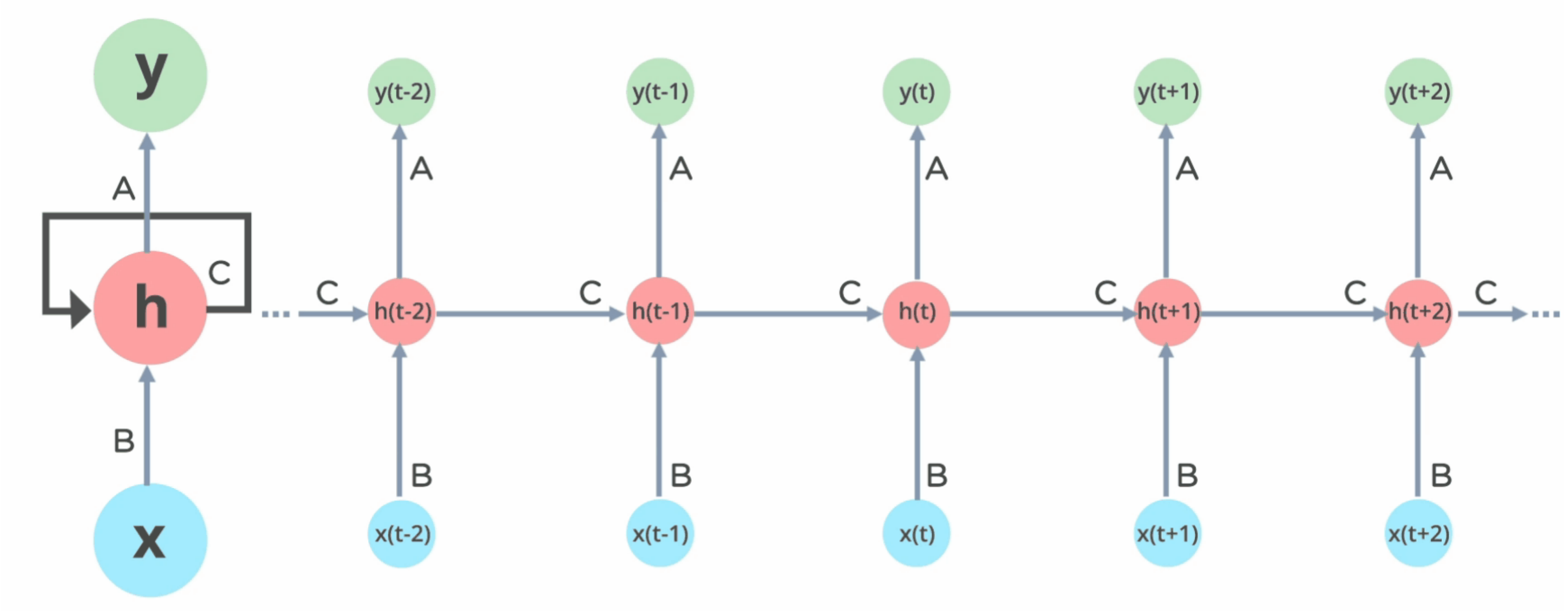


Image

Convolved Feature

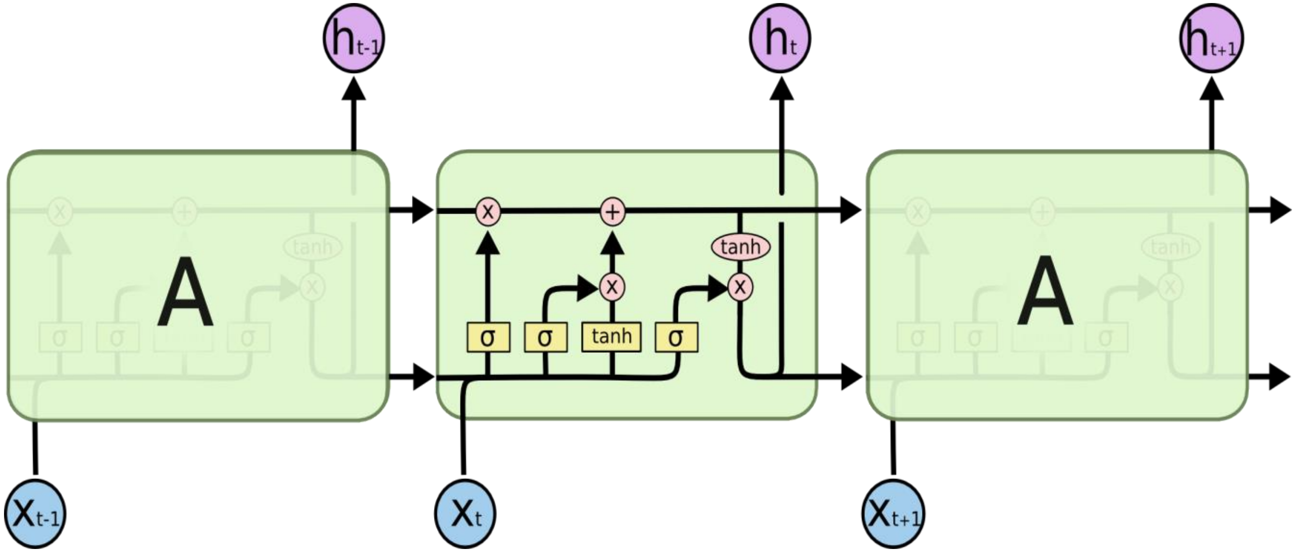


Recurrent Neural Networks (RNN)



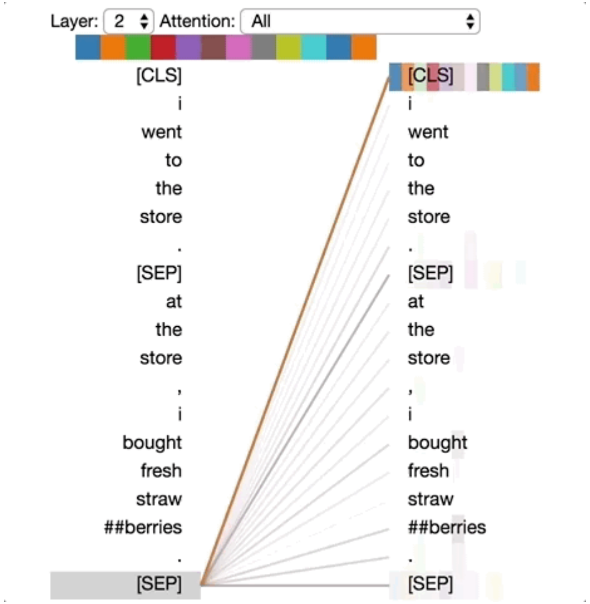
Long Short-Term Memory (LSTM)

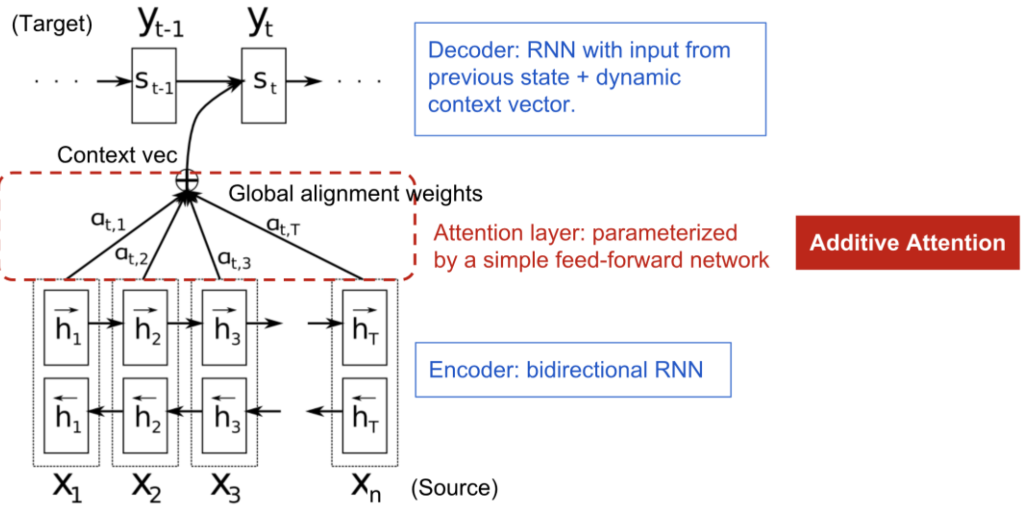
- Avoiding gradient explosion and vanishing



Attention Mechanism

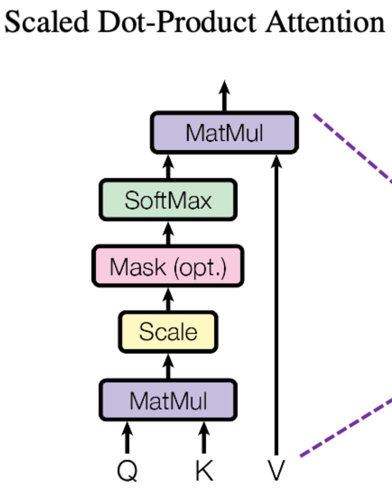
- Born for machine translation [Bahdanau et al. 2013]

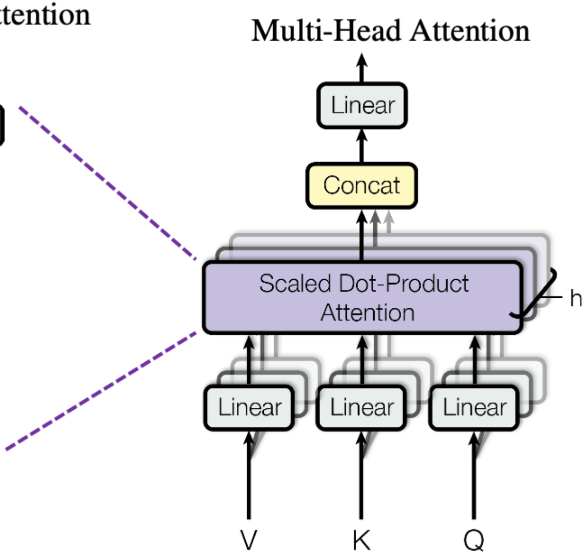


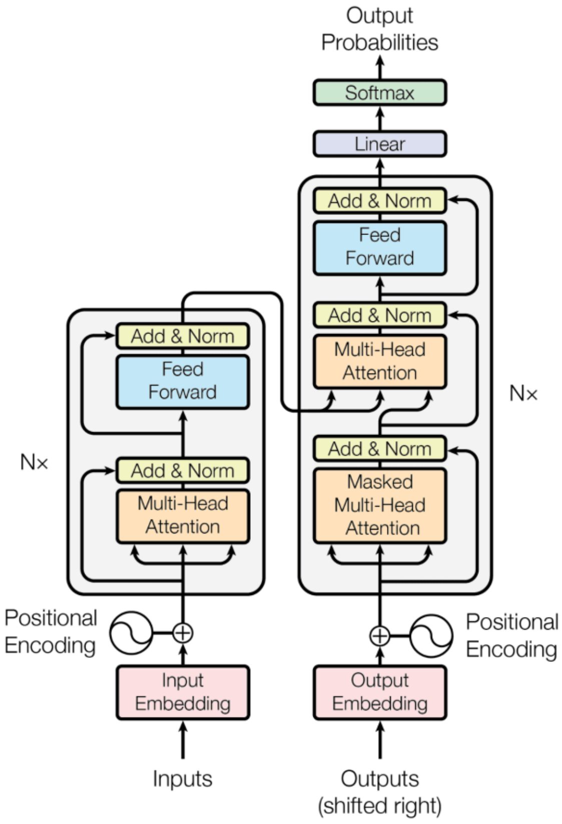


Transformers (Self-Attention)

- Attention is all you need

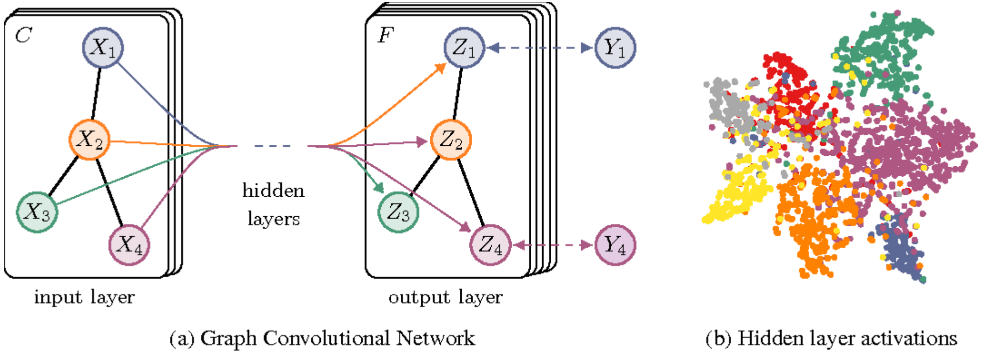


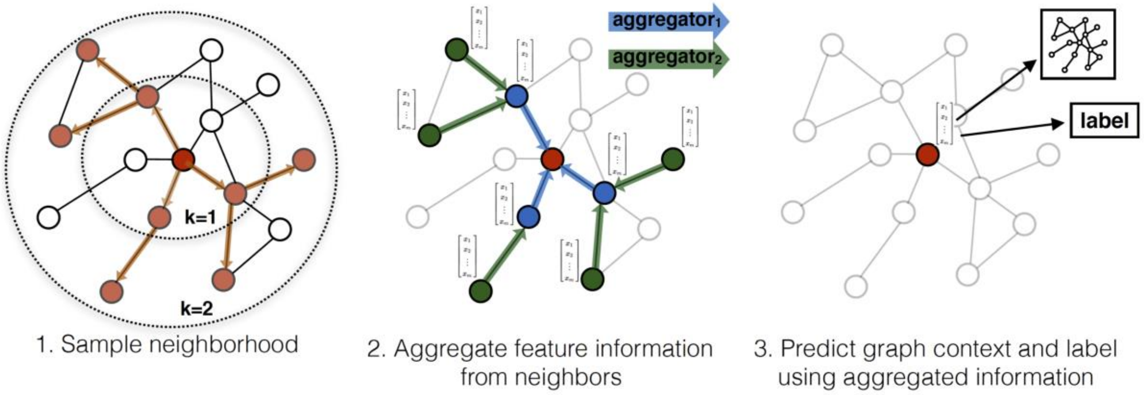




Graph Neural Networks (GNN)

- Spectral-based GNNs
- Spatial-based GNNs



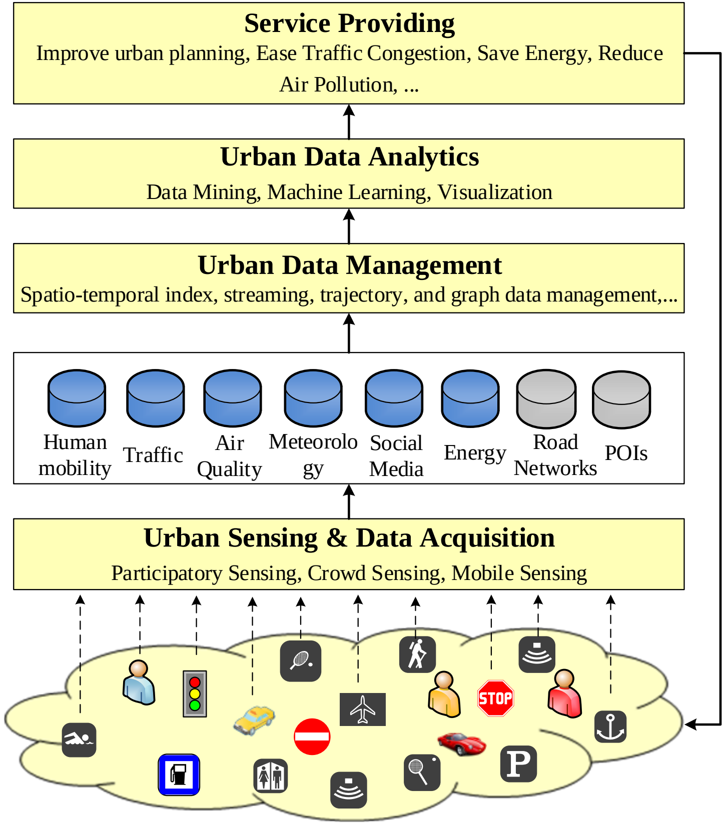


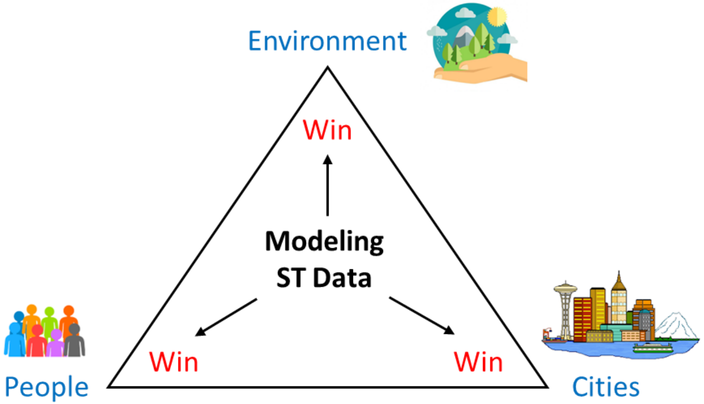
Objectives of this Course

To introduce

- The challenges of urban data sensing and processing
  - Biased distribution
  - Sparsity
  - Missing entries
  - Resource allocation
- The general framework of urban data sensing
  - Sensor-centric
  - Active crowd sourcing

Framework





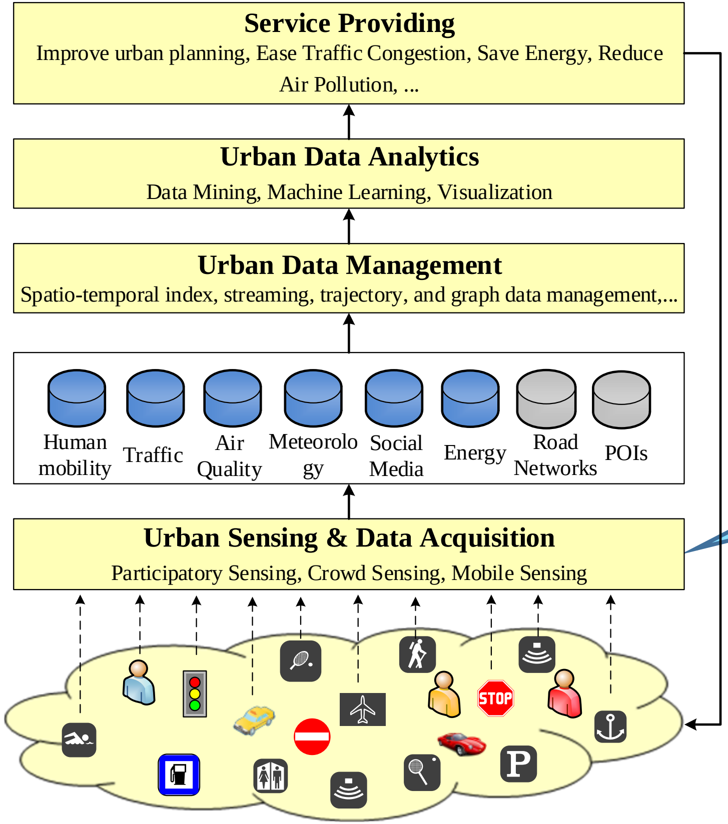
Tackle the Big challenges

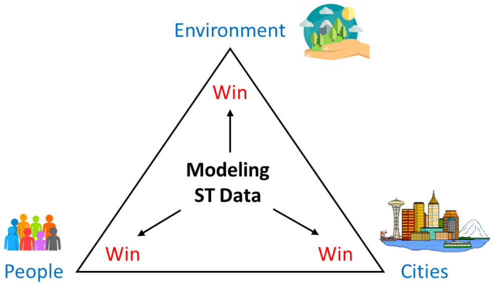
in Big cities

using Big data!

Urban Computing: concepts, methodologies, and applications. Zheng, Y., et al. *ACM transactions on Intelligent Systems and Technology*.

1st Stage: Urban Sensing & Data Acquision



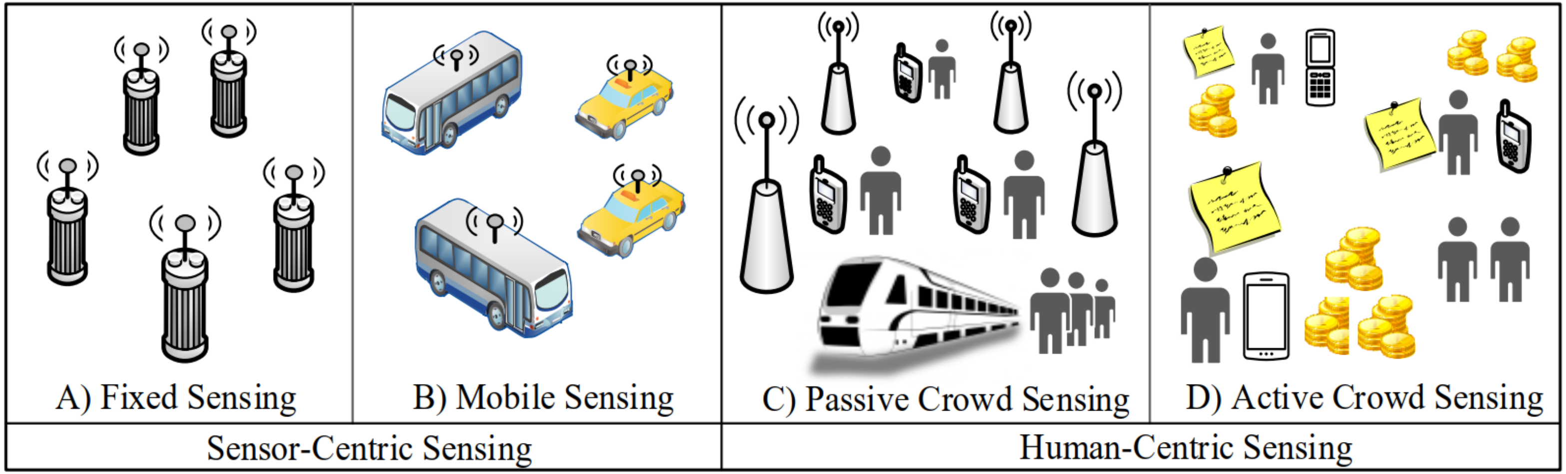


Collecting urban data through
- Sensor-centric sensing
- Human-centric sensing

Urban Sensing & Data Acquisition
Crowd Sensing: Passive Sensing, Participatory Sensing, Task Assignment...
Fixed sensing: Sensor Deployment, Location Selection for Sensors...

Spatio-Temporal Sensing Modes

- Sensor-centric sensing
  - Fixed (location) sensing
  - Mobile (objects) sensing
- Human-centric sensing
  - Passive crowd sensing
  - Active crowd sensing



Objectives of this Course

To introduce

- The challenges of urban data sensing and processing
  - Biased distribution
  - Sparsity
  - Missing entries
  - Resource allocation
- The general framework of urban data sensing
  - Sensor-centric
  - Active crowd sourcing

Challenges of Urban Sensing

Biased distribution

Data sparsity

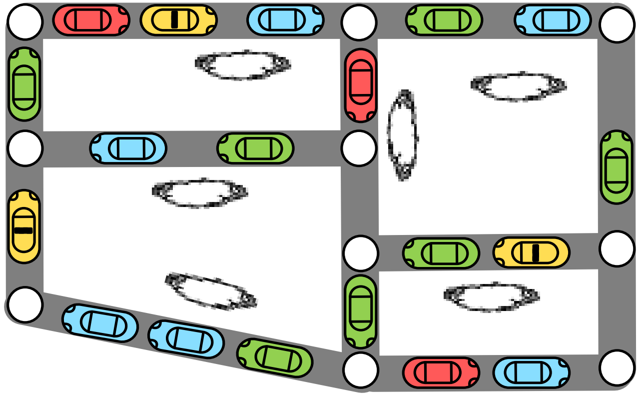
Data missing

Resource deployment

Inferring Gas Consumption and Pollution Emission of Vehicles throughout a City

KDD 2014

Jingbo Shang, et al. Inferring Gas Consumption and Pollution Emission of Vehicles throughout a City, KDD 2014



Questions

- How many liters of gas have been consumed by the vehicles, in the entire city, in the past one hour?
- What is the volume of PM2.5 that has been generated accordingly?

UNITED NATIONS GLOBAL PULSE

GLOBAL PULSE

HOME
RESEARCH
LABS
BLOG
CHALLENGES
LAB DIARIES
ABOUT
CONTACT

PULSE LAB DIARIES

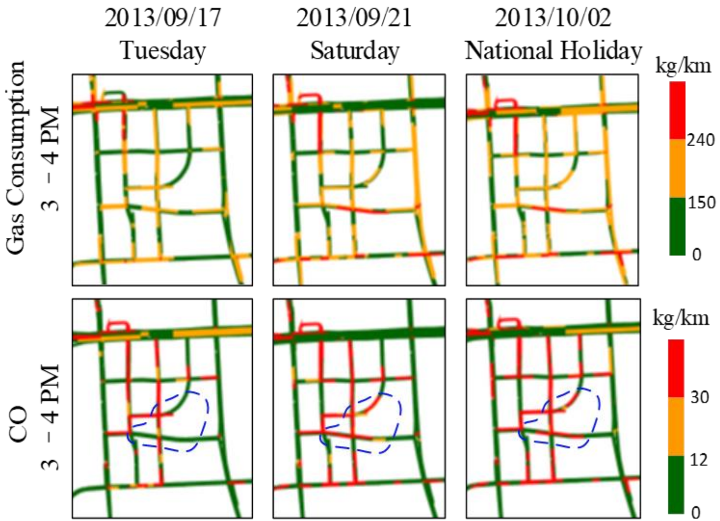
Research Bites: “Inferring Gas Consumption and Pollution Emission of Vehicles throughout a City”

Aug 28, 2014

As part of our "Research Bites" series, in which we ask **data science researchers to spend five minutes telling us in their own words about their work**, and opportunities for practical applicability in the context of sustainable development or humanitarian action, today we hear from researcher Yu Zheng from the Beijing lab of Microsoft Research. Yu is a co-author of this work along with researchers Jingbo Shang, Wenzhu Tong, Eric Chang and Yong Yu.

Goals

- Estimate the gas consumption and vehicle emissions
  - on arbitrary road segment
  - at any time intervals
  - using GPS trajectories of a sample of vehicles



Model

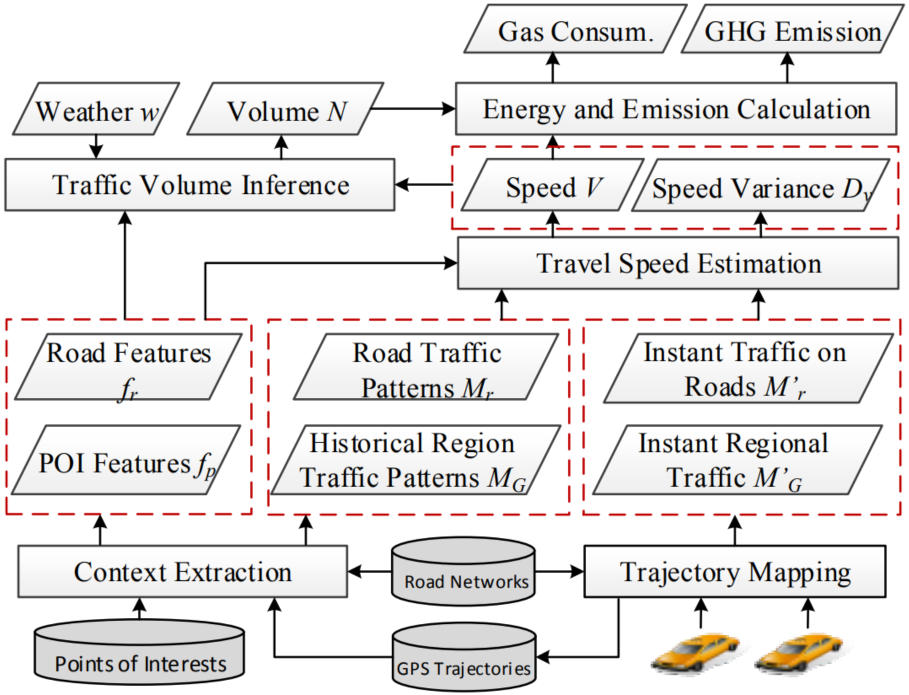


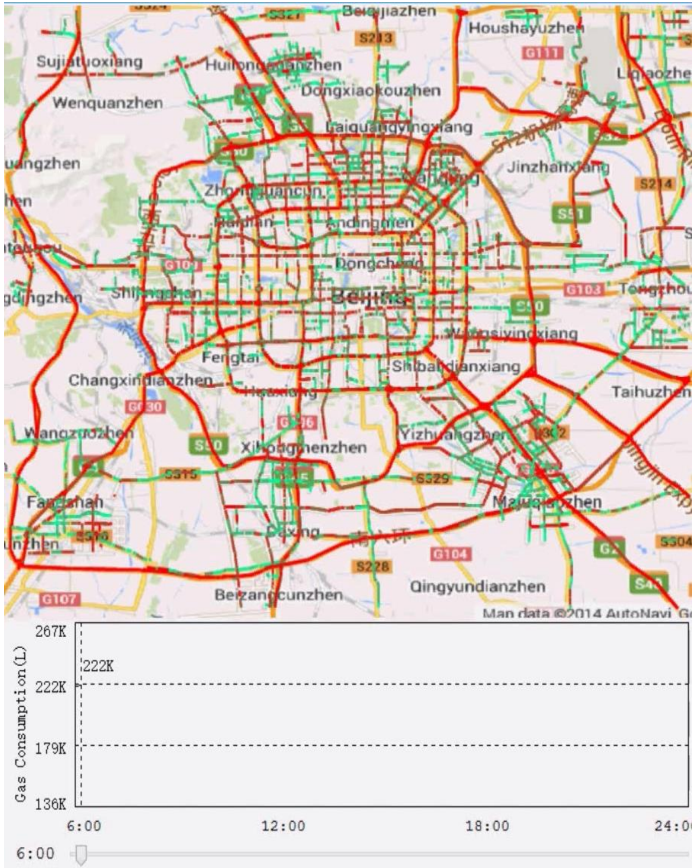
Figure 2. Framework of our method

$$EF = (a + cv + ev^2)/(1 + bv + dv^2).$$

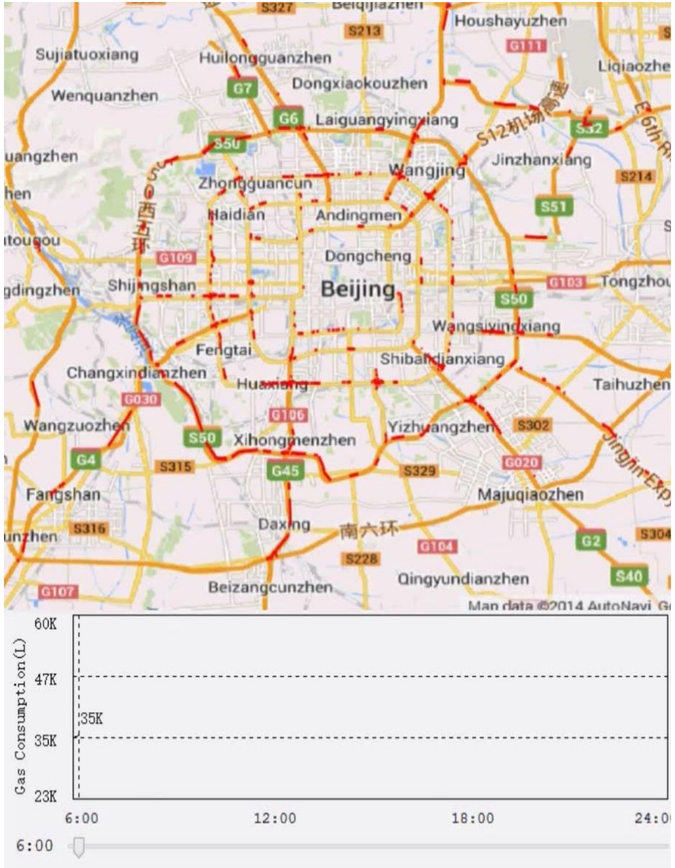
Real-Time Gas Consumption

2 7 20 40L

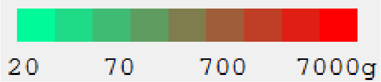
Citywide



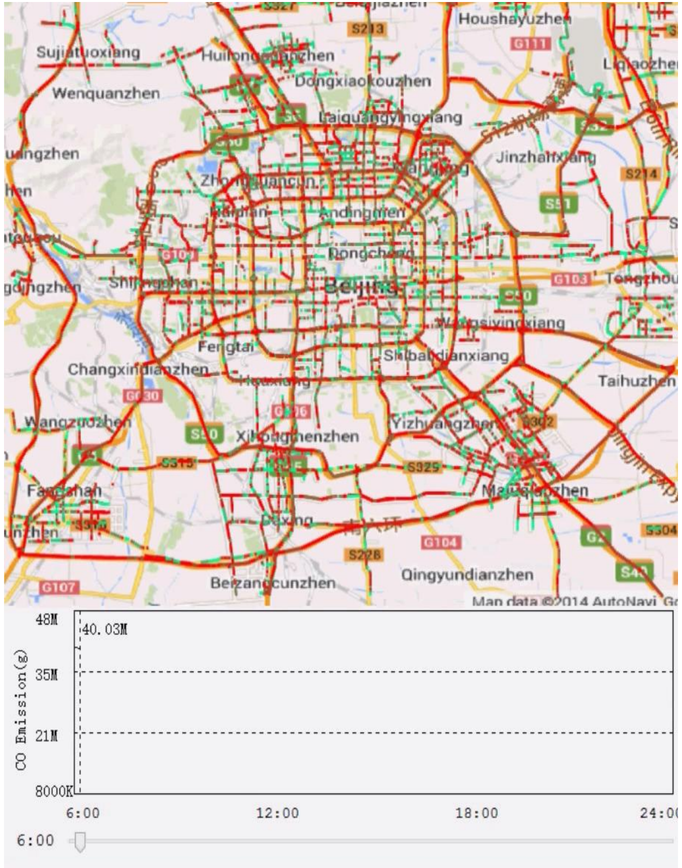
Top 3%



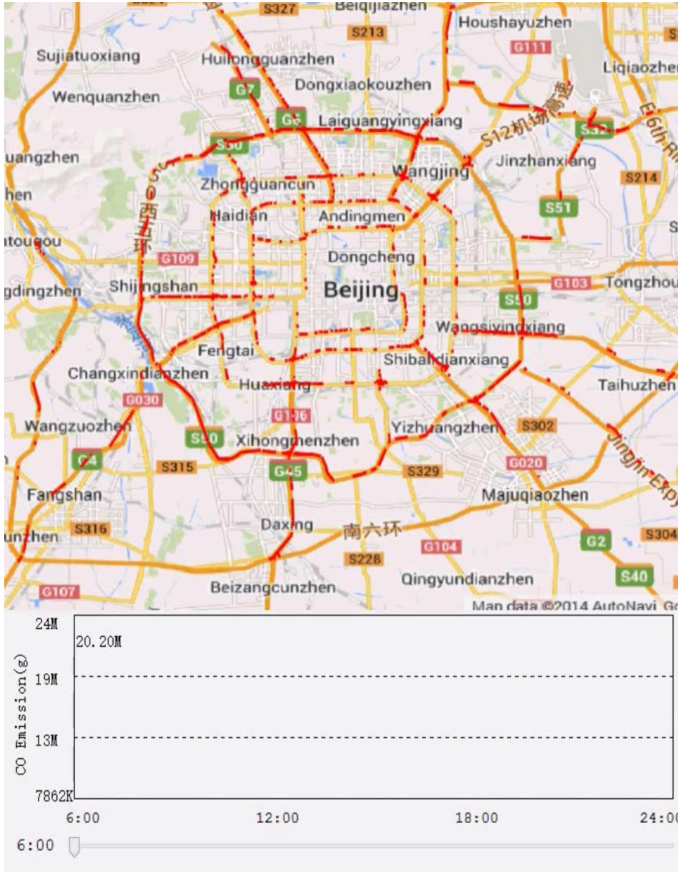
Real-Time CO Emission



Citywide



Top 4%



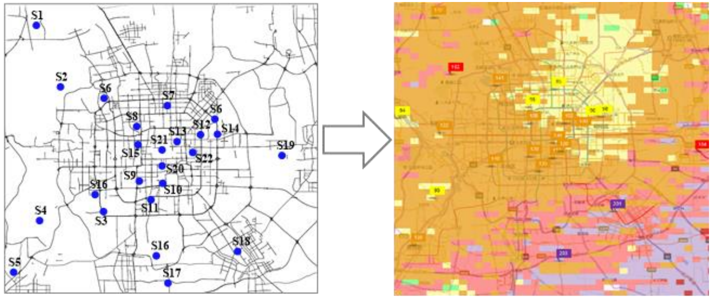
A Sample of Data → An Entire Dataset

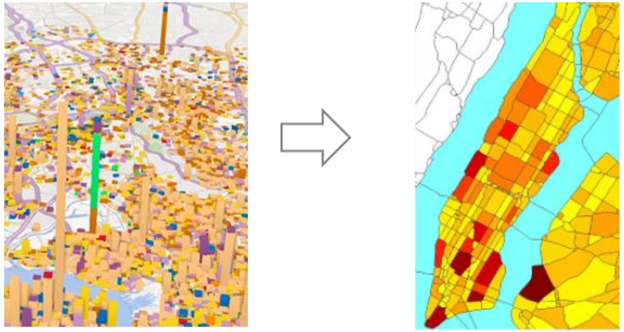
- Biased distribution

Taxi flow Entire traffic flow

Check in data Citywide human mobility

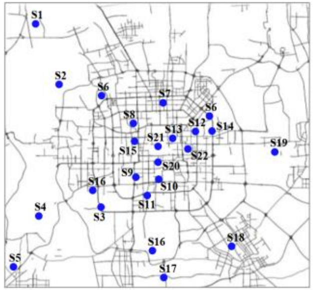
- Data missing and sparsity





Challenges of Urban Sensing

Biased distribution



Data sparsity

Data missing

Resource deployment

When Urban Air Meets Big Data

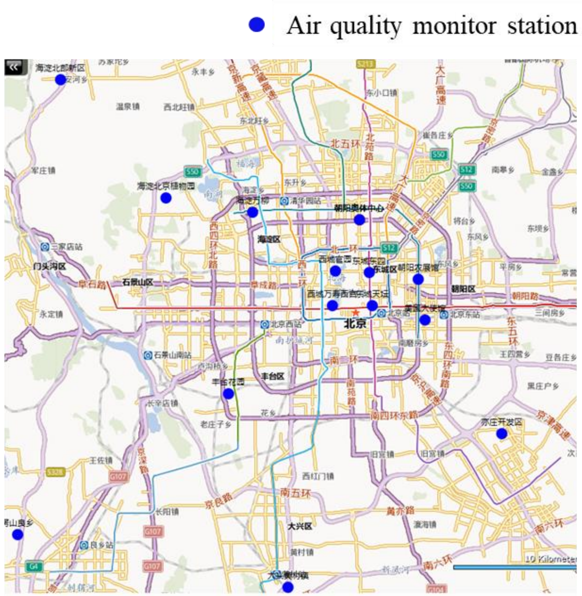
KDD 2013

Yu Zheng, et al. U-Air: When Urban Air Quality Inference Meets Big Data, KDD 2013

Background

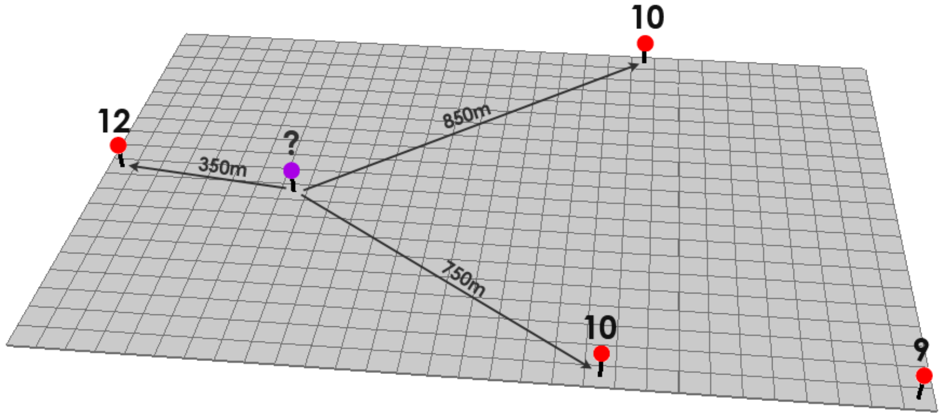
- Air quality
  - NO2, SO2
  - Aerosols: PM2.5, PM10
- Why it matters
  - Healthcare
  - Pollution control and dispersal
- Reality
  - Building a measurement station is not easy
  - A limited number of stations (poor coverage)

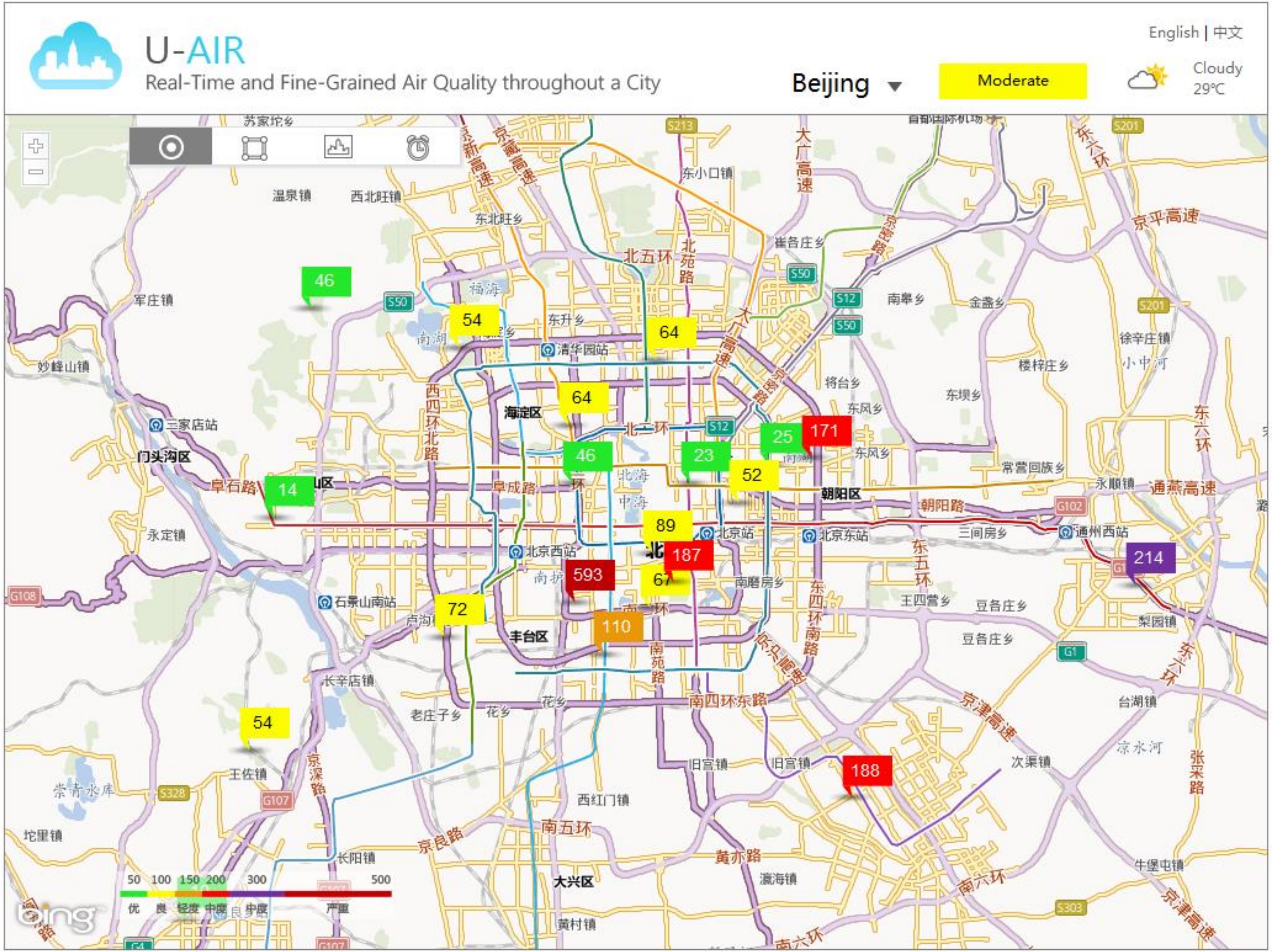
Beijing only has 15 air quality monitor stations in its urban areas (50kmx40km)



Traditional Solutions

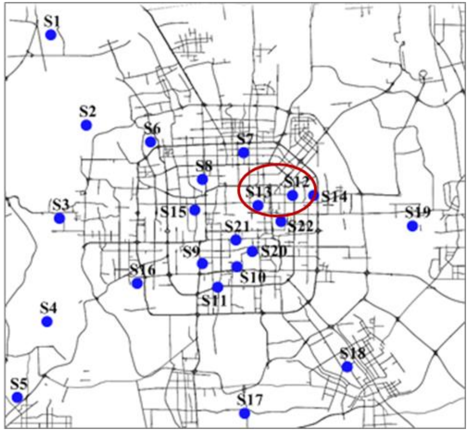
- K-nearest neighbors (KNN)
- Inverse distance weighting (IDW)
- Kriging
- Random forest
- SVR

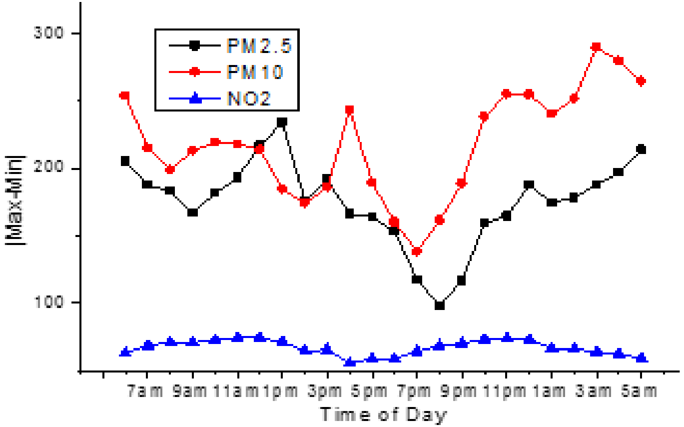



U-AIR
Real-Time and Fine-Grained Air Quality throughout a City
English | 中文
Beijing
Moderate
Cloudy
29℃
50 100 150 200 300 500
优 良 轻度 中度 重度 严重

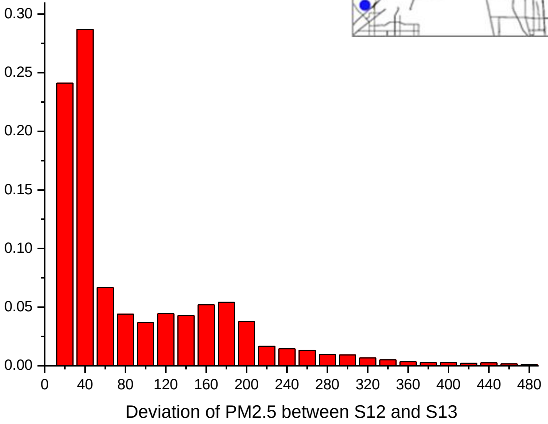
Challenges

- Air quality varies by locations non-linearly
  - Affected by many factors
  - Weathers, traffic, land use…
- Subtle to model with a clear formula





A) Beijing (8/24/2012 - 3/8/2013)



We do not really know the air quality of a location without a monitoring station!

Inferring **Real-Time** and **Fine-Grained** air quality throughout a city using **Big Data**

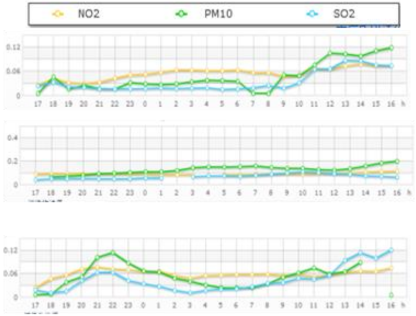
Meteorology

Traffic

Human Mobility

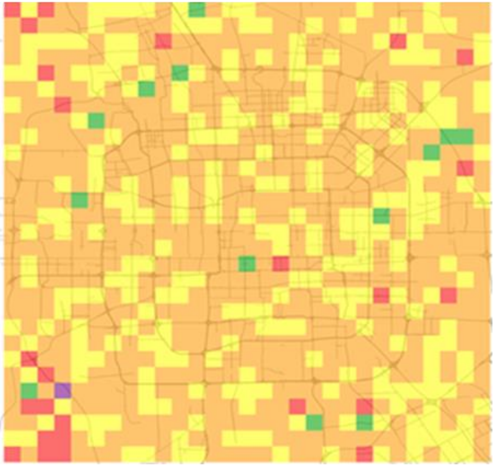
POIs

Road networks

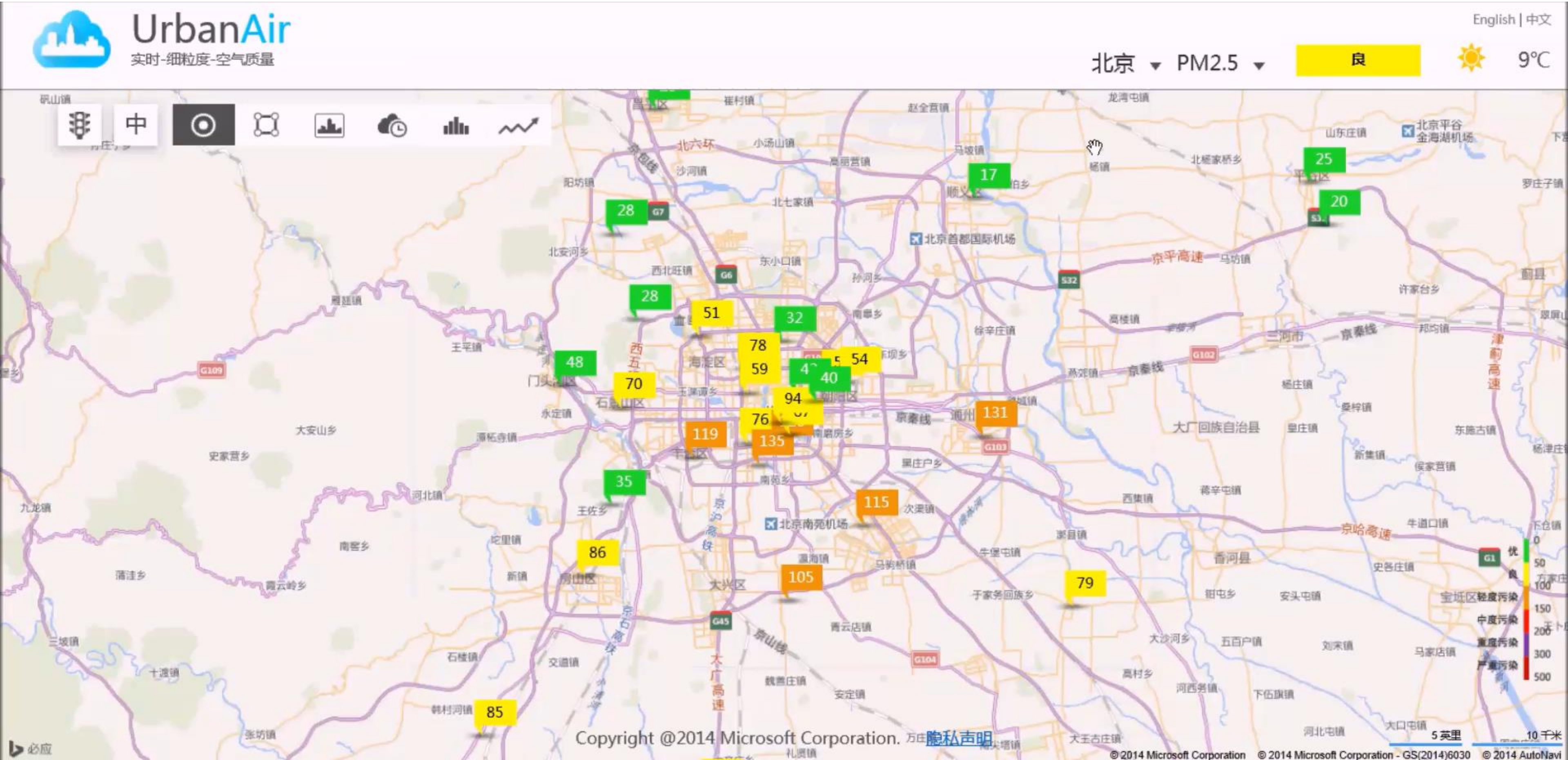


Historical air quality data

Real-time air quality reports

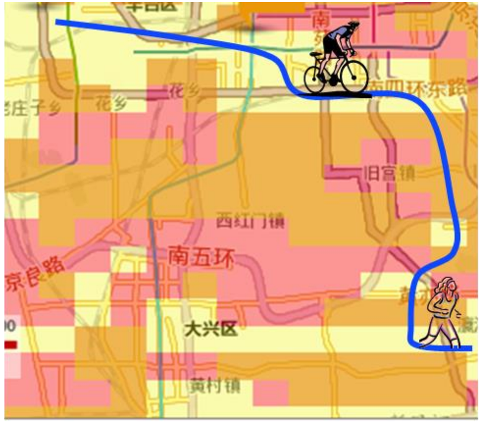


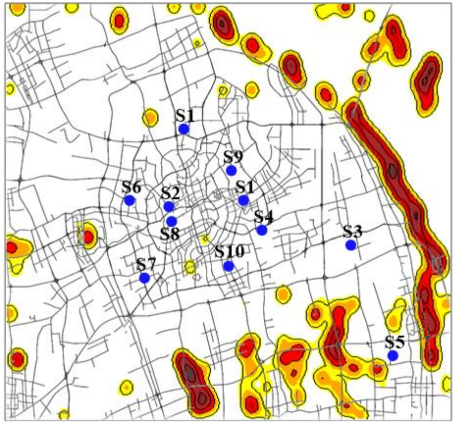
http://urbanair.msra.cn



Applications

- Location-based air quality awareness
  - Fine-grained pollution alert
  - Routing based on air quality
- Identify candidate locations for setup new monitoring stations
- A step towards identifying the root cause of air pollution





Difficulties

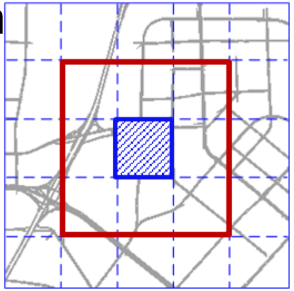
- How to identify features from each kind of data source

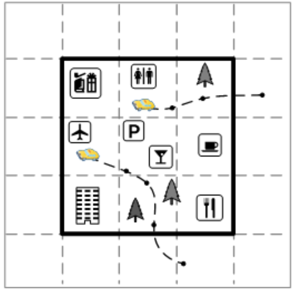
- Incorporate multiple heterogeneous data sources into a learning model
  - Spatially-related data: POIs, road networks
  - Temporally-related data: traffic, meteorology, human mobility

- Data sparseness (little training data)
  - Limited number of stations
  - Many places to infer

Methodology Overview

- Partition a city into disjoint grids
- Extract features for each grid from its affecting region
  - Meteorological features
  - Traffic features
  - Human mobility features
  - POI features
  - Road network features





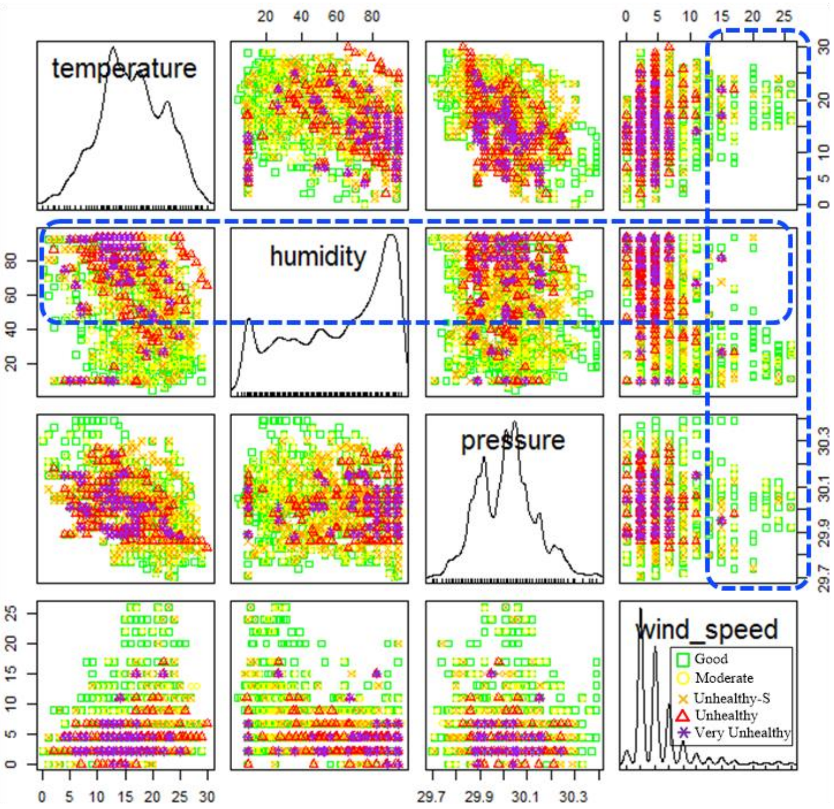
- Co-training-based semi-supervised learning model for each pollutant
  - Predict the AQI labels
  - Data sparsity
  - Two classifiers

AQI	Values Levels of Health Concern	Colors
0-50	Good (G)	Green
51-100	Moderate (M)	Yellow
101-150	Unhealthy for sensitive groups (U-S)	Orange
151-200	Unhealthy (U)	Red
201-300	Very unhealthy (VU)	Purple
301-500	Hazardous (H)	Maroon

Meteorological Features: F_m

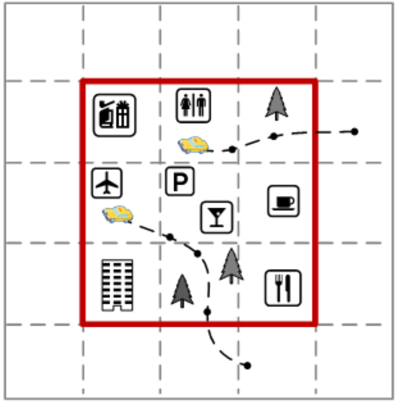
- Rainy, Sunny, Cloudy, Foggy
- Wind speed
- Temperature
- Humidity
- Barometer pressure

AQI of PM_{10}
August to Dec. 2012 in Beijing

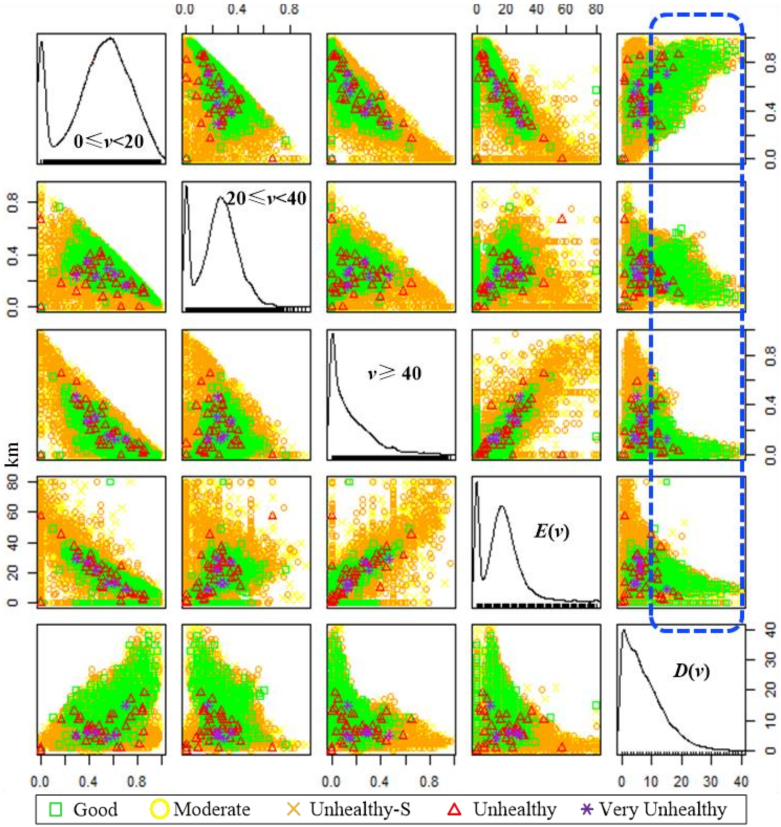


Traffic Features: F_t

- Distribution of speed by time: F(v)
- Expectation of speed: E(V)
- Standard deviation of Speed: D

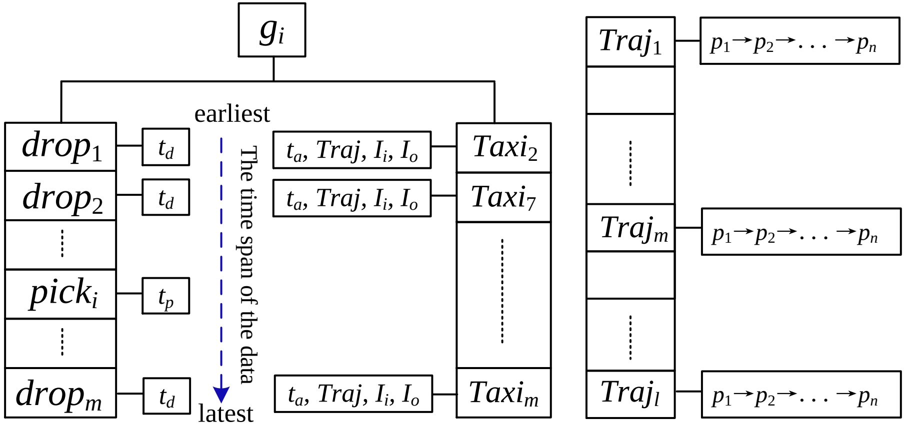


GPS trajectories generated by over 30,000 taxis
From August to Dec. 2012 in Beijing



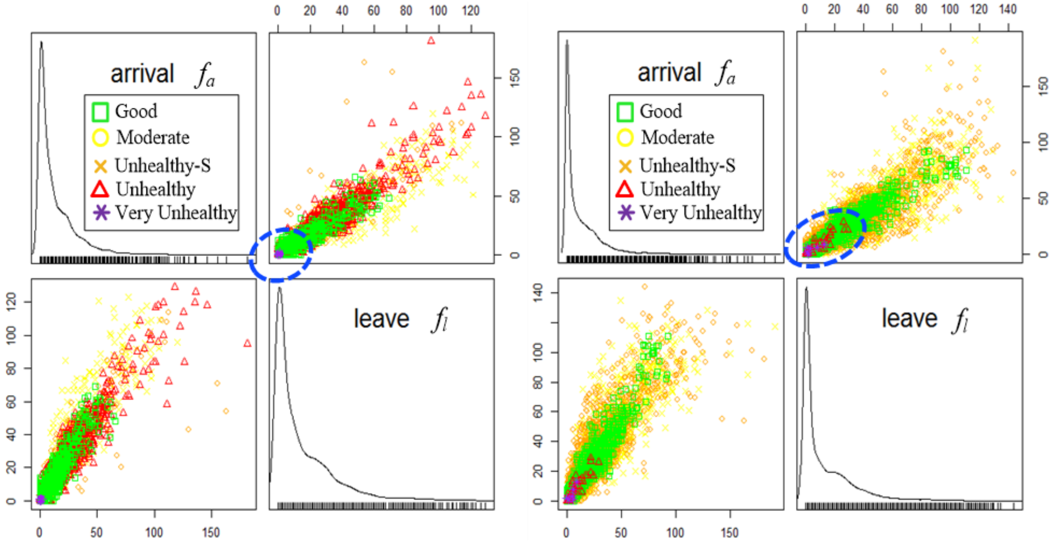
Extracting Traffic Features

- Offline spatio-temporal indexing
  - t_a: arrival time
  - Traj: trajectory ID
  - l_i: the index of the first GPS point (in the trajectory) entering a grid
  - l_o: the index of the last GPS point (in the trajectory) leaving a grid



Human Mobility Features: F_h

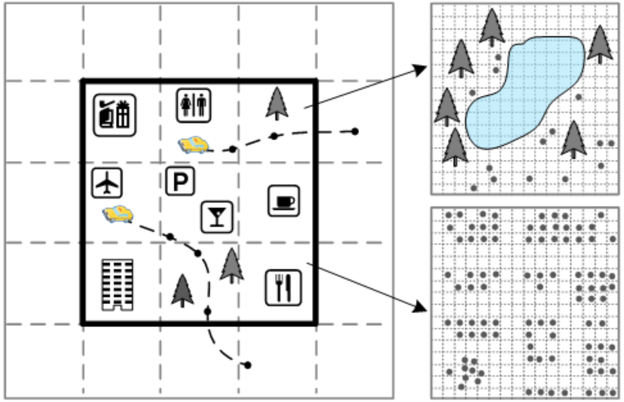
- Human mobility implies
  - Traffic flow
  - Land use of a location
  - Function of a region (like residential or business areas)
- Features:
  - Number of arrivals f_a and leavings f_l



A) AQI of PM_{10} B) AQI of NO_2

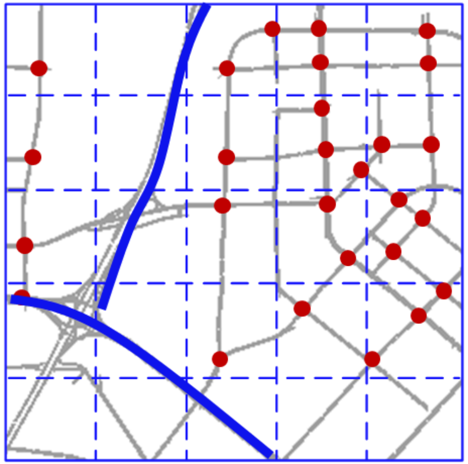
POI Features: F_p

- Why POI
  - Indicate the land use/function of the region
  - the traffic patterns in the region
- Features
  - Distribution of POIs over categories
  - Portion of vacant places
  - The changes in the number of POIs
    - ➢ Factories, shopping malls,
    - ➢ hotel and real estates
    - ➢ Parks, decoration and furniture markets



C_1: Vehicle Services (gas stations, repair)	C_7: Sports
C_2: Transportation spots	C_8: Parks
C_3: Factories	C_9: Culture & education
C_4: Decoration and furniture markets	C_{10}: Entertainment
C_5: Food and beverage	C_{11}:Companies
C_6: Shopping malls and Supermarkets	C_{12}:Hotels and real estates

Road Network Features: F_r

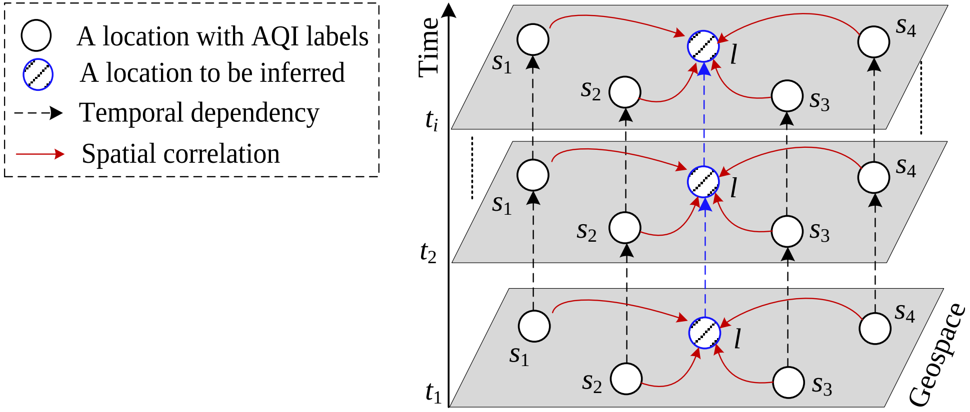


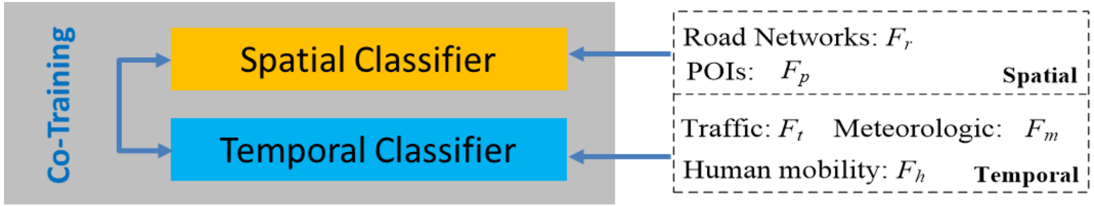
- Why road networks
  - Have a strong correlation with traffic flows
  - A good complementary of traffic modeling

- Features:
  - Total length of highways f_h
  - Total length of other (low-level) road segments f_r
  - The number of intersections f_s in the grid's affecting region

Semi-Supervised Learning Model

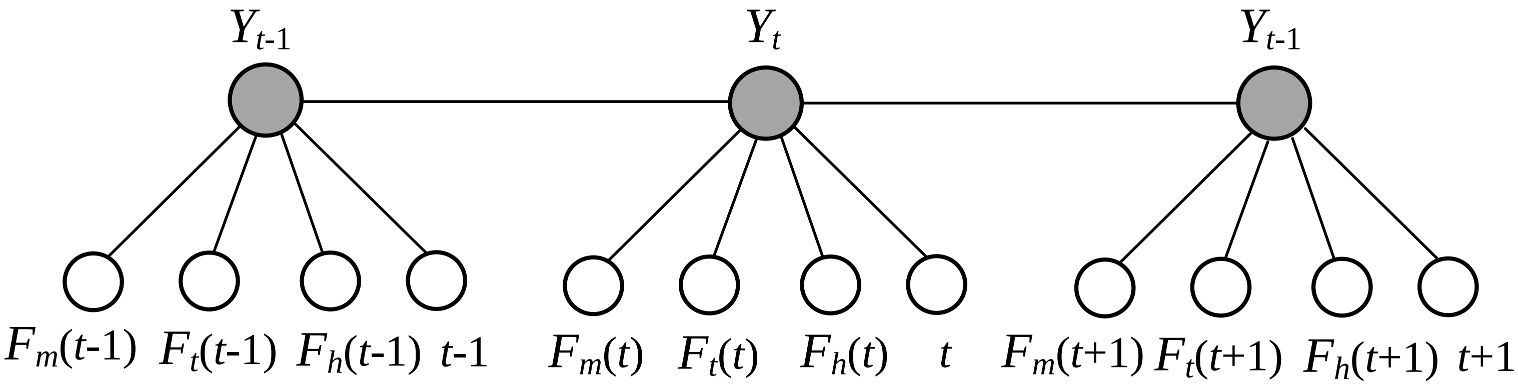
- Philosophy of the model
  - States of air quality
    - ➢ Temporal dependency in a location
    - ➢ Geo-correlation between locations
  - Generation of air pollutants
    - ➢ Emission from a location
    - ➢ Propagation among locations
  - Two sets of features
    - ➢ Spatially-related
    - ➢ Temporally-related





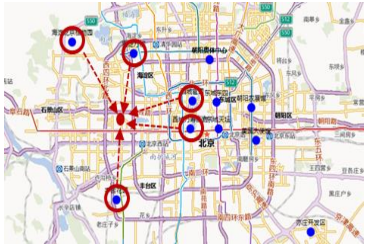
Co-Training-Based Learning Model

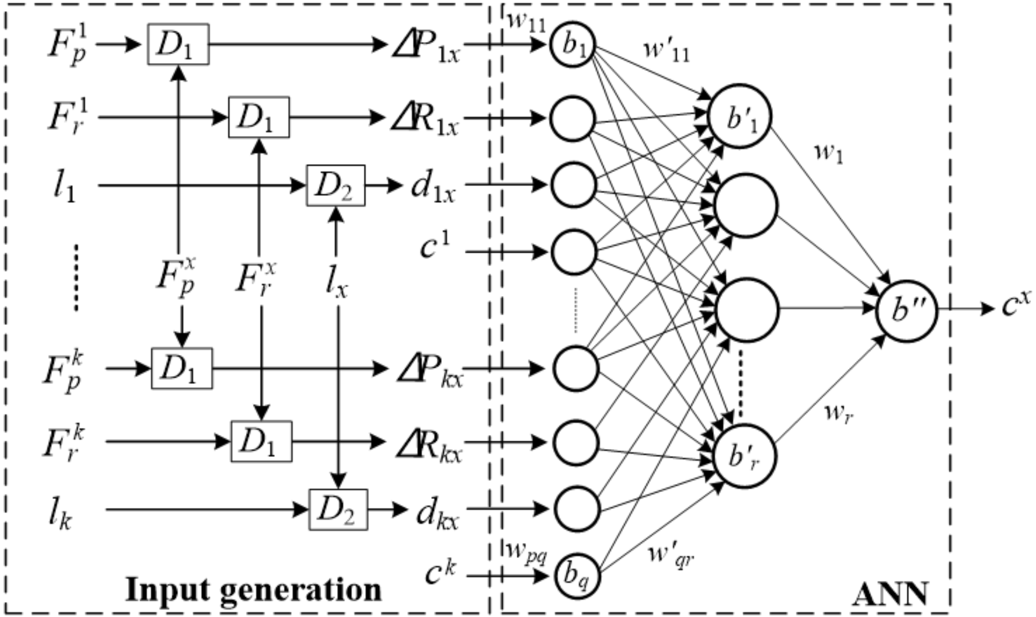
- Temporal classifier
  - Model the temporal dependency of the air quality in a location
  - Using temporally related features
  - Based on a Linear-Chain Conditional Random Field (CRF)



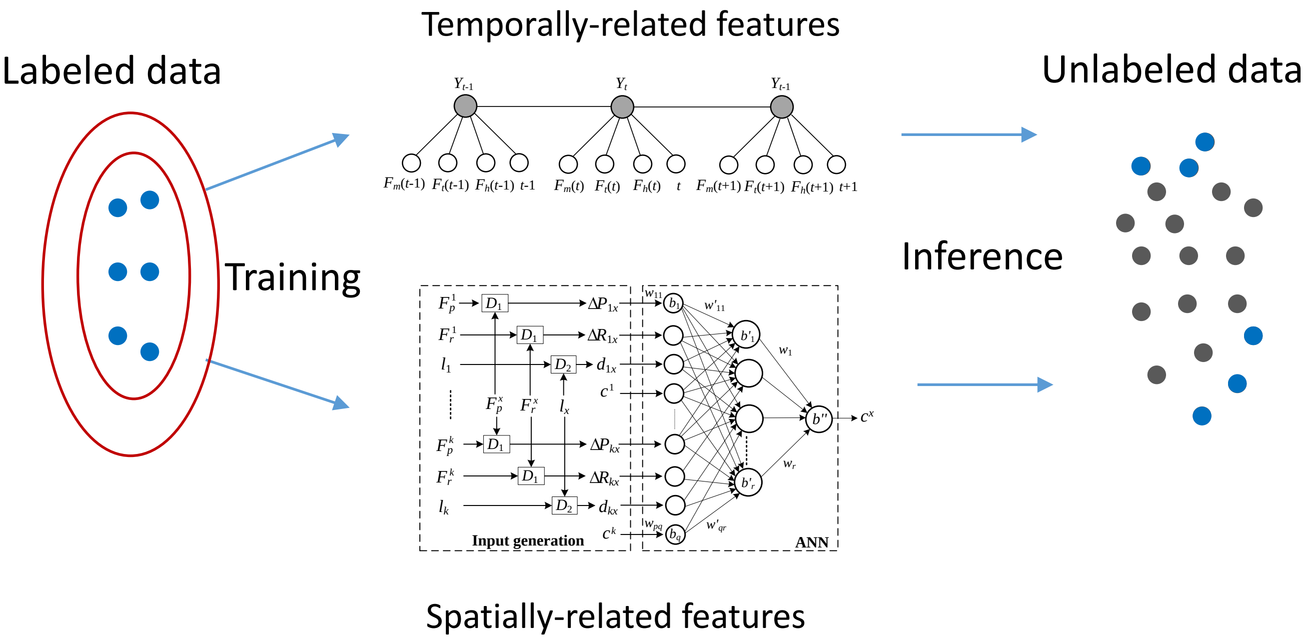
Co-Training-Based Learning Model

- Spatial classifier
  - Model the spatial correlation between AQI of different locations
  - Using spatially-related features
  - Based on a BP neural network
- Input generation
  - Select n stations to pair with
  - Perform m rounds



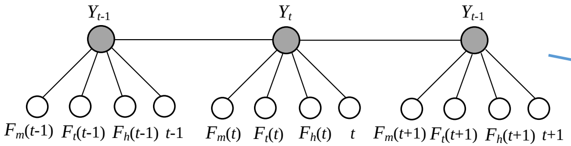


Learning Process



Inference Process

Temporally-related features



Inference

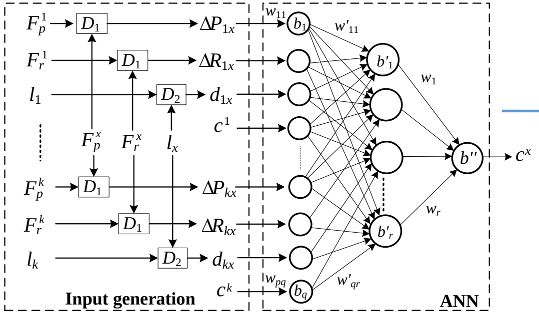
Inference

$$< p_{c1}, p_{c2}, \ldots, p_{cn} >$$

$$\times$$

$$c = \arg_{c_i \in C} Max(p_{ci} \times p'_{ci})$$

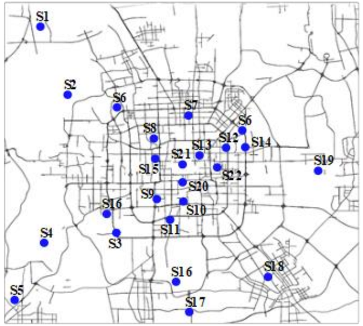
$$< p'_{c1}, p'_{c2}, \ldots, p'_{cn} >$$



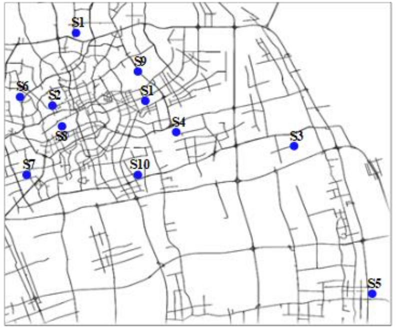
Spatially-related features

Datasets

Data sources		Beijing	Shanghai	Shenzhen	Wuhan
POI	2012 Q1	271,634	321,529	107,061	102,467
	2012 Q3	272,109	317,829	107,171	104,634
Road	#.Segments	162,246	171,191	45,231	38,477
	Highways	1,497km	1,963km	256km	1,193km
	Roads	18,525km	25,530km	6,100km	9,691km
	#. Intersec.	49,981	70,293	32,112	25,359
AQI	#. Station	22	10	9	10
	Hours	23,300	8,588	6,489	6,741
	Time spans	8/24/2012-3/8/2013	1/19/2013-3/8/2013	2/4/2013-3/8/2013	2/4/2013-3/8/2013
Urban Size (grids)		50×50km (2500)	50×50km (2500)	57×45km(2565)	45×25km (1165)

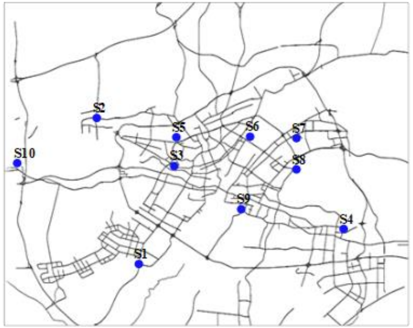


A) Beijing



B) Shanghai

C) Shenzhen



D) Wuhan

Evaluation

- Ground Truth
  - Remove a station
  - Cross cities
- Baselines
  - Linear and Gaussian Interpolations
  - Classical Dispersion Model
  - Decision Tree (DT):
  - CRF-ALL
  - ANN-ALL

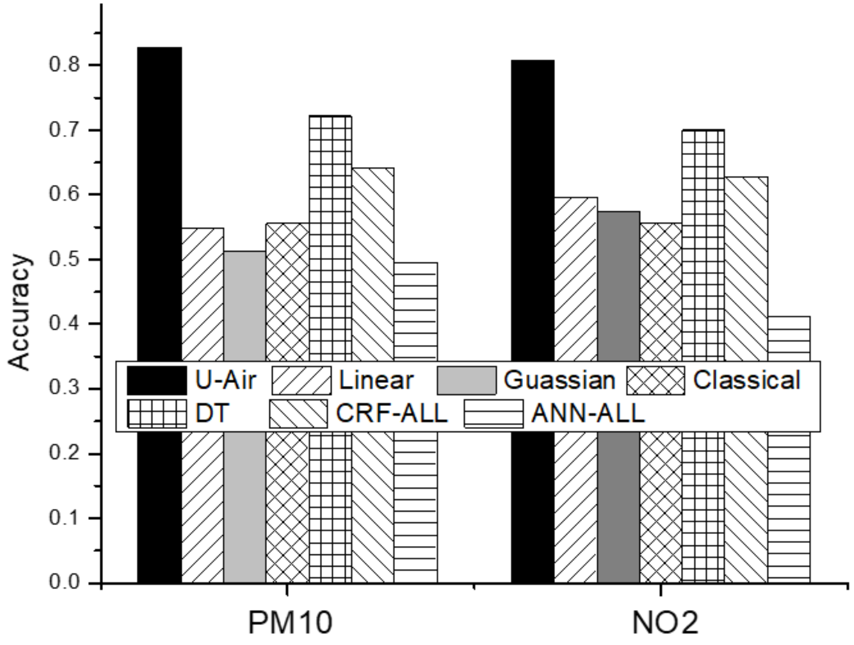
Evaluation

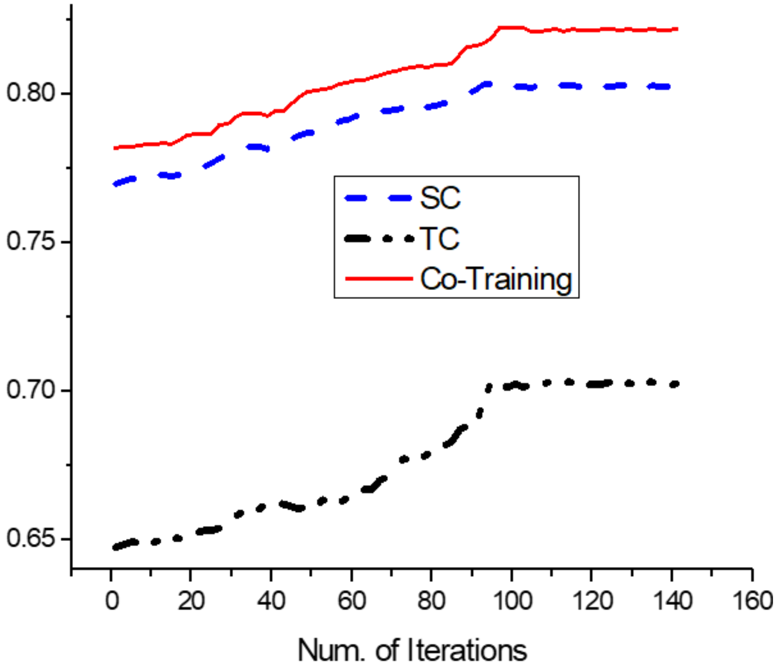
- Does every kind of feature count?

	PM10		NO2	
Features	Precision	Recall	Precision	Recall
F_m	0.572	0.514	0.477	0.454
F_t	0.341	0.36	0.371	0.35
F_h	0.327	0.364	0.411	0.483
F_p+F_r	0.441	0.443	0.307	0.354
F_m+F_t	0.664	0.675	0.634	0.635
F_m+F_t+F_p+F_r	0.731	0.734	0.701	0.691
F_m+F_t+F_p+F_r+F_h	0.773	0.754	0.723	0.704

Evaluation

- Overall performance of the co-training





Evaluation

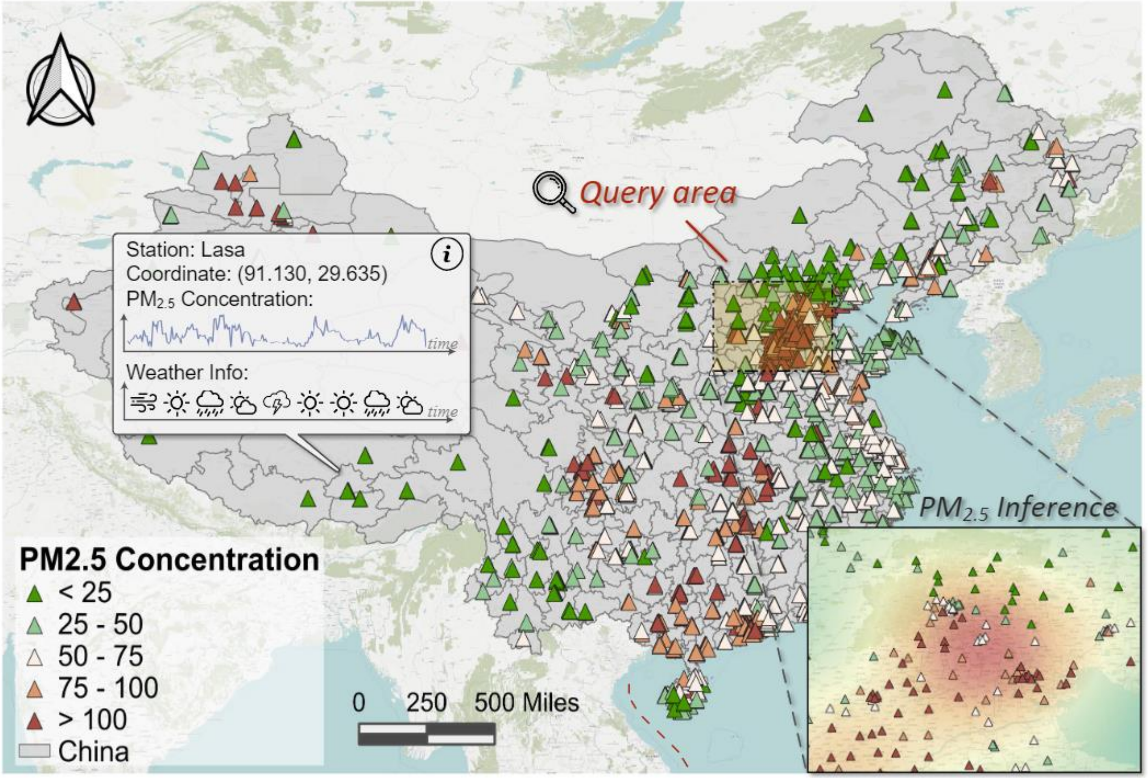
- Efficiency study
- Inferring the AQIs for entire Beijing in 5 minutes

Procedures		Time(ms)	Procedures		Time(ms)
Feature extraction (per grid)	F_t & F_h	53.2	Inference (per grid)	SC	21.5
	F_p	28.8		TC	13.1
	F_r	14.4	Total		131

Conclusion

- Infer fine-grained air quality with
  - Real-time and historical air quality readings from existing stations
  - Other data sources: meteorology, POIs, road network, human mobility, and traffic condition
- Co-Training-based semi-supervised learning approach
  - Deal with data sparsity by learning from unlabeled data
  - Model the spatial correlation among the air quality of different locations
  - Model the temporal dependency of the air quality in a location
- Results
  - 0.82 with traffic data (co-training)
  - 0.76 if only using spatial classifier

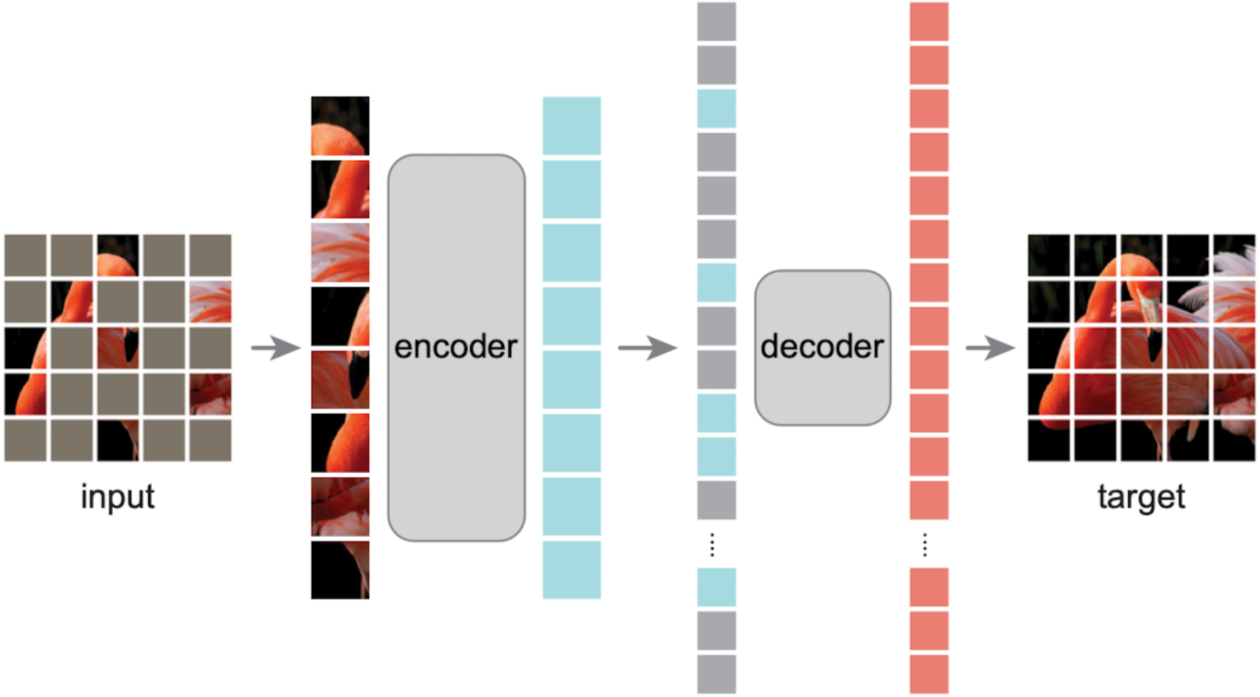
Extending to a Nationwide Scale



Wang et al. [AirRadar: Inferring Nationwide Air Quality in China with Deep Neural Networks](), Under Review

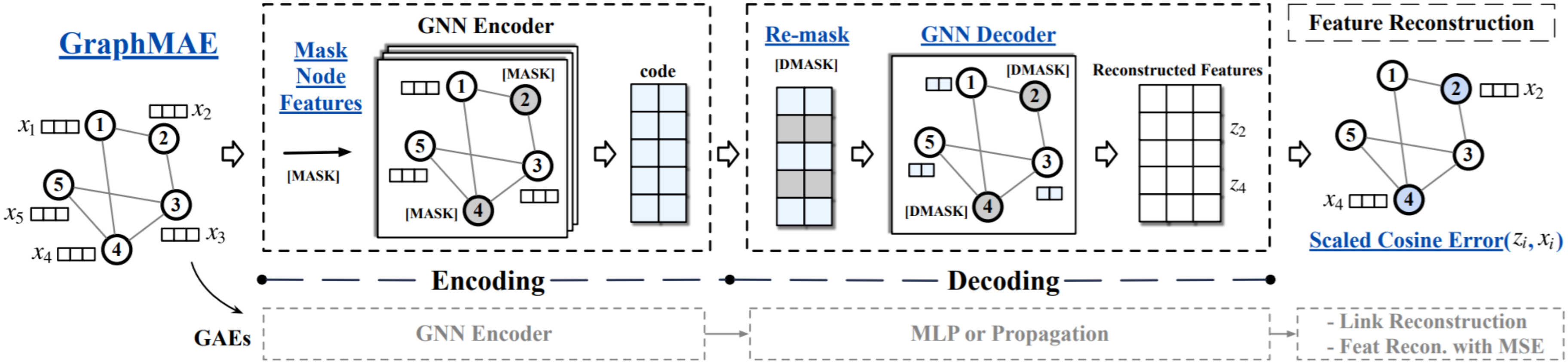
Related Work

- Masked AutoEncoder (MAE) [He et al. 2021]



Related Work

- GraphMAE [He et al. 2021]



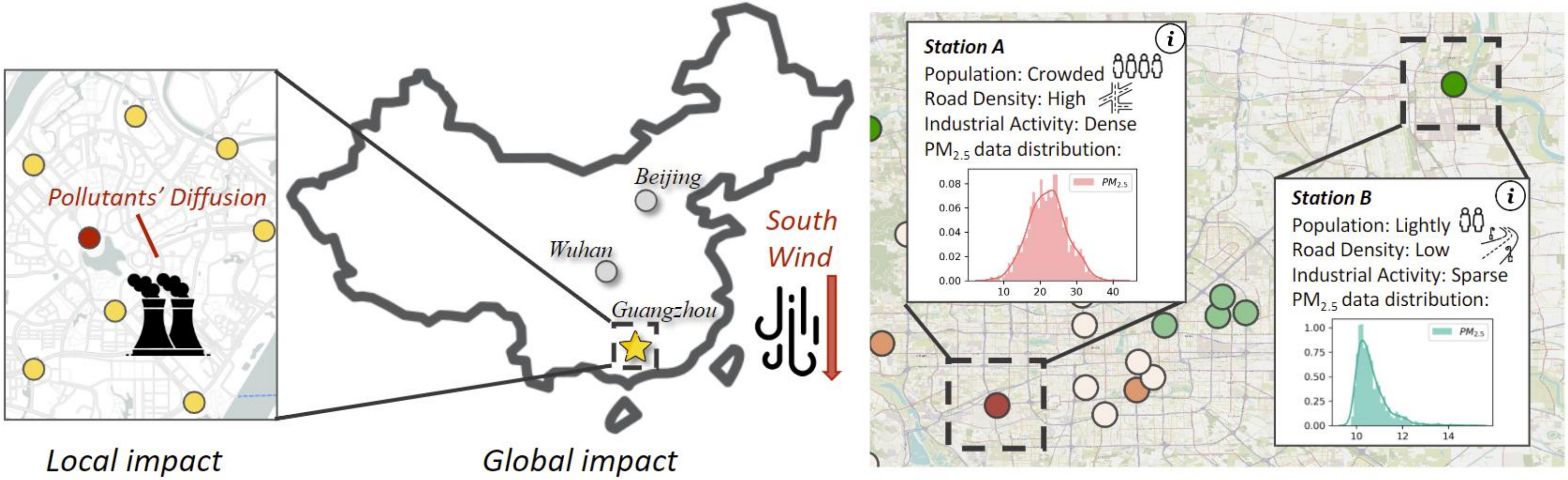
Results of GraphMAE

Table III. Model comparison on the nationwide dataset. The parameter count, denoted as #Param, is in the order of million (M). The symbol Δ represents the reduction in MAE compared to GraphMAE. The mask ratio represents the proportion of unobserved nodes to all nodes.

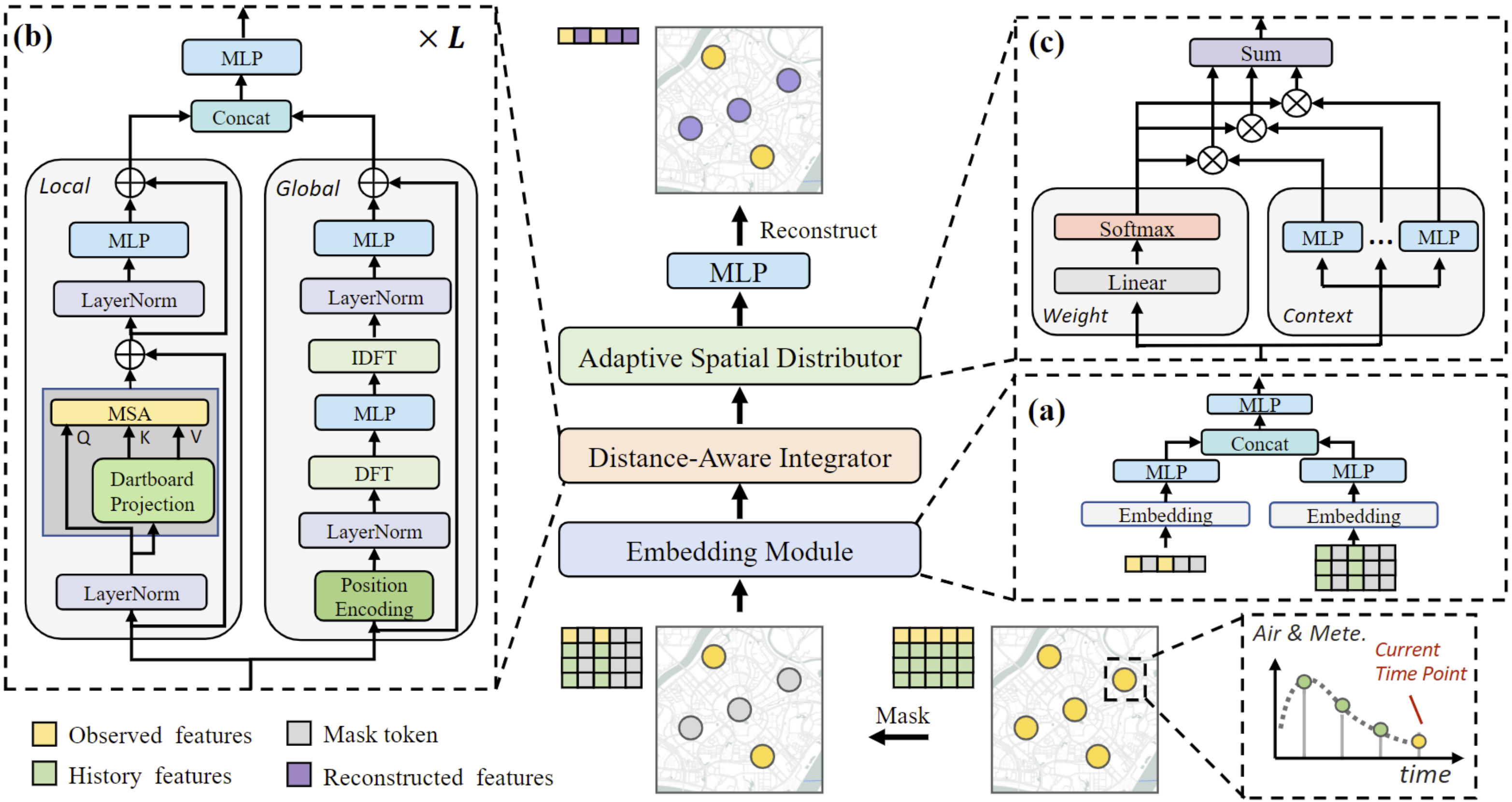
Model	Year	#Param(M)	Mask Ratio = 25%				Mask Ratio = 50%				Mask Ratio = 75%			
			MAE	Δ	RMSE	MAPE	MAE	Δ	RMSE	MAPE	MAE	Δ	RMSE	MAPE
KNN	1967	-	30.50	+146.0%	65.40	1.36	30.25	+145.5%	72.23	0.71	34.07	+194.0%	74.55	0.64
RF	2001	-	29.22	+135.6%	68.95	0.76	29.71	+141.2%	71.61	0.75	29.82	+157.3%	70.99	0.74
MCAM	2021	0.408	23.94	+93.1%	36.25	0.95	25.01	+103.0%	37.94	0.92	25.19	+117.3%	37.82	1.04
SGNP	2019	0.114	23.60	+90.3%	37.58	0.83	24.06	+95.3%	37.08	0.93	21.68	+87.1%	33.68	0.84
STGNP	2022	0.108	23.21	+87.2%	38.13	0.62	21.95	+78.2%	37.13	0.67	19.58	+68.9%	31.95	0.69
VAE	2013	0.011	28.49	+129.8%	67.11	0.94	28.92	+134.7%	69.67	0.94	29.00	+150.2%	69.11	0.93
GAE	2016	0.073	12.63	+1.9%	23.80	0.46	12.78	+3.7%	24.11	0.46	12.57	+8.5%	23.73	0.46
GraphMAE	2022	0.073	12.40	-	23.20	0.46	12.32	-	23.11	0.46	11.59	-	21.51	0.43

Challenges

- Local and global impact
- Spatial distribution shift



Framework

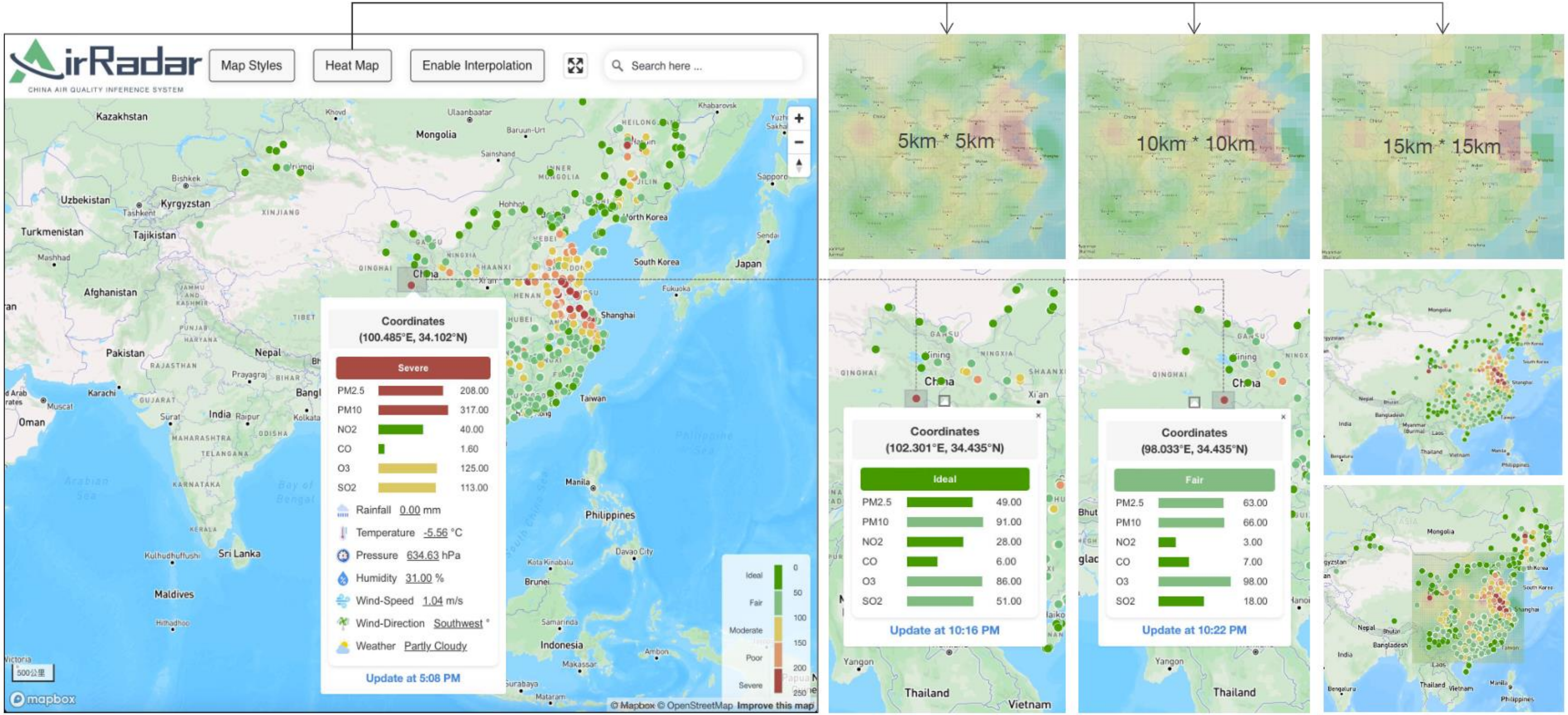


(b)
× L
MLP
Concat
Local
MLP
LayerNorm
MSA
Q
K
V
Dartboard Projection
LayerNorm
Global
MLP
LayerNorm
IDFT
MLP
DFT
LayerNorm
Position Encoding
Reconstruct
MLP
Adaptive Spatial Distributor
Distance-Aware Integrator
Embedding Module
Mask
(c)
Sum
Softmax
Linear
Weight
MLP
MLP
Context
(a)
MLP
Concat
MLP
MLP
Embedding
Embedding
Air & Mete.
Current Time Point
time
Observed features
History features
Mask token
Reconstructed features

Results

Model	Year	#Param(M)	Mask Ratio = 25%				Mask Ratio = 50%				Mask Ratio = 75%			
			MAE	Δ	RMSE	MAPE	MAE	Δ	RMSE	MAPE	MAE	Δ	RMSE	MAPE
KNN	1967	-	30.50	+146.0%	65.40	1.36	30.25	+145.5%	72.23	0.71	34.07	+194.0%	74.55	0.64
RF	2001	-	29.22	+135.6%	68.95	0.76	29.71	+141.2%	71.61	0.75	29.82	+157.3%	70.99	0.74
MCAM	2021	0.408	23.94	+93.1%	36.25	0.95	25.01	+103.0%	37.94	0.92	25.19	+117.3%	37.82	1.04
SGNP	2019	0.114	23.60	+90.3%	37.58	0.83	24.06	+95.3%	37.08	0.93	21.68	+87.1%	33.68	0.84
STGNP	2022	0.108	23.21	+87.2%	38.13	0.62	21.95	+78.2%	37.13	0.67	19.58	+68.9%	31.95	0.69
VAE	2013	0.011	28.49	+129.8%	67.11	0.94	28.92	+134.7%	69.67	0.94	29.00	+150.2%	69.11	0.93
GAE	2016	0.073	12.63	+1.9%	23.80	0.46	12.78	+3.7%	24.11	0.46	12.57	+8.5%	23.73	0.46
GraphMAE	2022	0.073	12.40	-	23.20	0.46	12.32	-	23.11	0.46	11.59	-	21.51	0.43
AirRadar	-	0.343	**6.41**	**-48.3%**	**12.60**	**0.24**	**6.79**	**-44.9%**	**12.90**	**0.26**	**8.11**	**-30.0%**	**14.84**	**0.29**

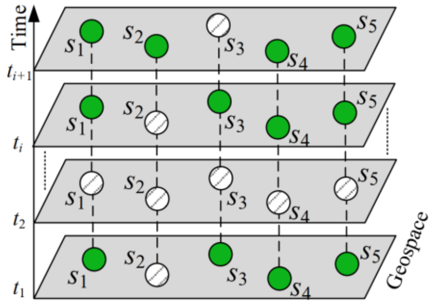
Demo System


AirRadar
CHINA AIR QUALITY INFERENCE SYSTEM
Map Styles
Heat Map
Enable Interpolation
Search here ...
Coordinates
(100.485°E, 34.102°N)
Severe
PM2.5 208.00
PM10 317.00
NO2 40.00
CO 1.60
O3 125.00
SO2 113.00
Rainfall 0.00 mm
Temperature -5.56 °C
Pressure 634.63 hPa
Humidity 31.00 %
Wind-Speed 1.04 m/s
Wind-Direction Southwest
Weather Partly Cloudy
Update at 5:08 PM
Ideal
Fair
Moderate
Poor
Severe
0
50
100
150
200
250
5km * 5km
10km * 10km
15km * 15km
Coordinates
(102.301°E, 34.435°N)
Ideal
PM2.5 49.00
PM10 91.00
NO2 28.00
CO 6.00
O3 86.00
SO2 51.00
Update at 10:16 PM
Coordinates
(98.033°E, 34.435°N)
Fair
PM2.5 63.00
PM10 66.00
NO2 3.00
CO 7.00
O3 98.00
SO2 18.00
Update at 10:22 PM
© Mapbox © OpenStreetMap Improve this map

Challenges of Urban Sensing

Biased distribution

Data sparsity



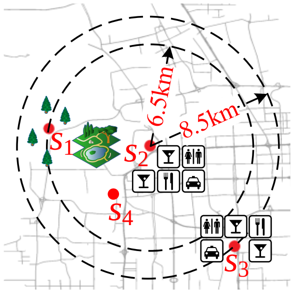
Data missing

Resource deployment

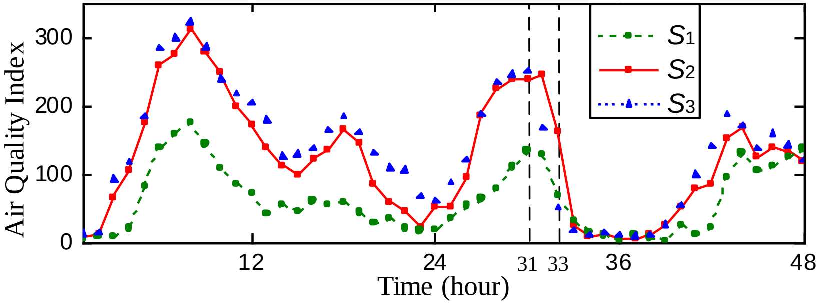
ST-MVL: Filling Missing Values in Geo-sensory Time Series Data

IJCAI 2016

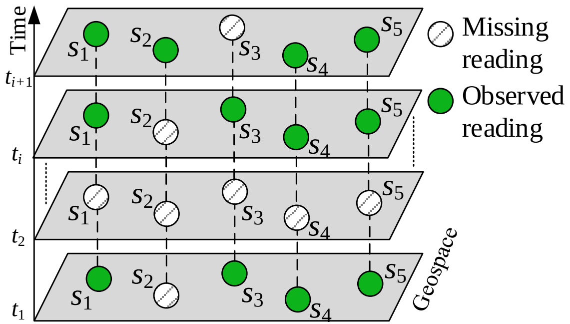
Yi et al. [ST-MVL: Filling Missing Values in Geo-sensory Time Series Data](), IJCAI 2016



A) Geo-location of sensors



B) Air quality index over time

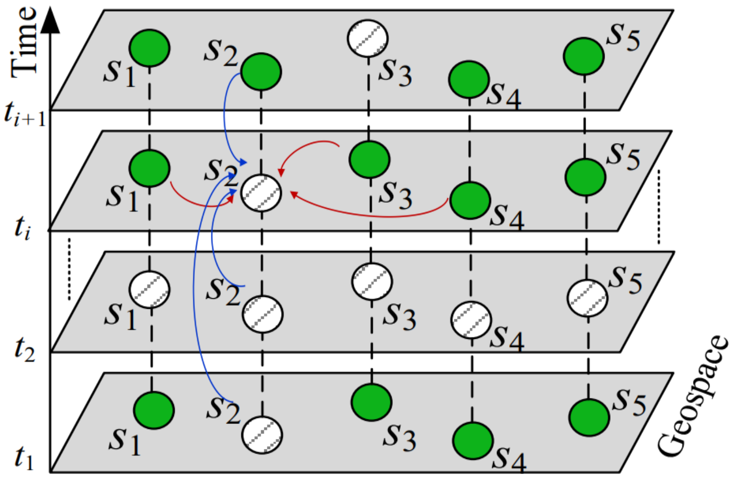


A) Missing situation

Filling Missing Values in Spatio-Temporal Data

- Data missing is a very common phenomenon in IOT data
  - Lost data that is supposed to have
  - Due to Communication or device errors
- Goal: Inferring missing values using:
  - Data of a sensor and
  - its neighborhoods
- A very fundamental problem
  - Important for monitoring
  - and further data analytics

	PM2.5	NO_2	Humidity	Wind Speed
Missing rate	13.3%	16.0%	21.5%	30.3%



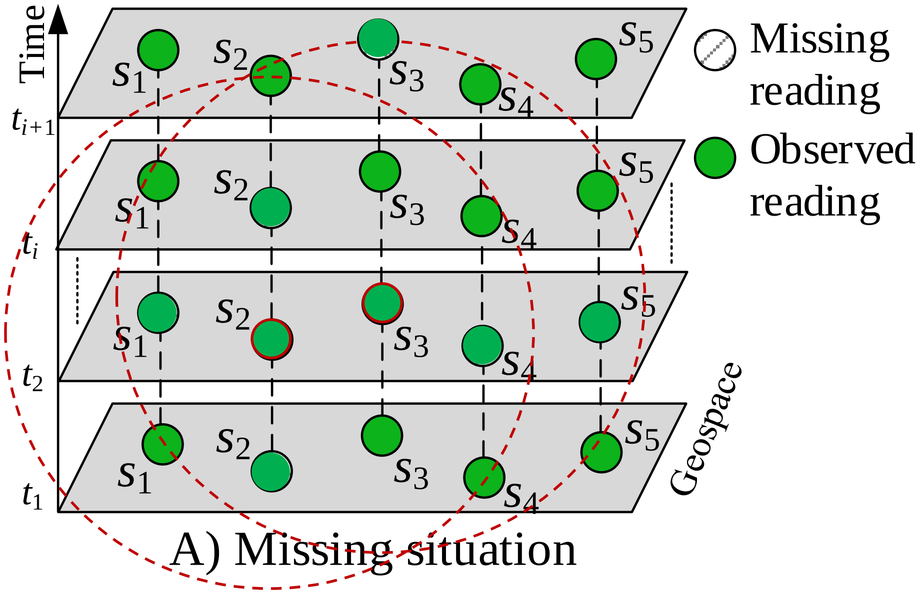
Xiuwen Yi et al., [ST-MVL: Filling Missing Values in Geo-Sensory Time Series Data](), IJCAI 2016

Filling Missing Values in Spatio-Temporal Data

- Goal: Inferring the values of those missing entries using collective information:
  - Data of a sensor and its neighborhoods

A very fundamental problem

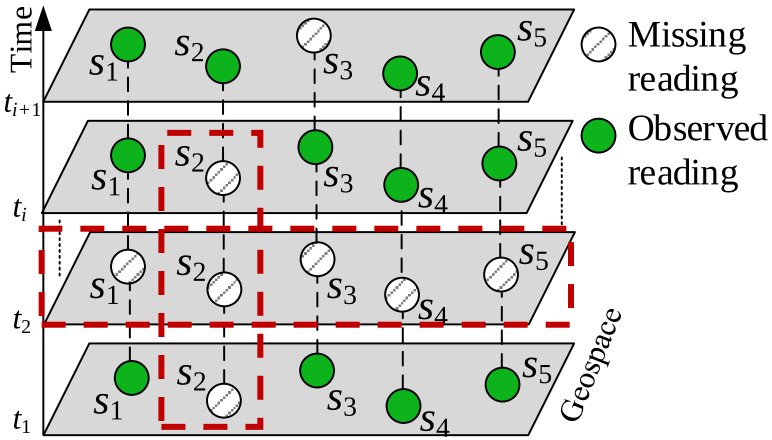
Important for monitoring and further data analytics



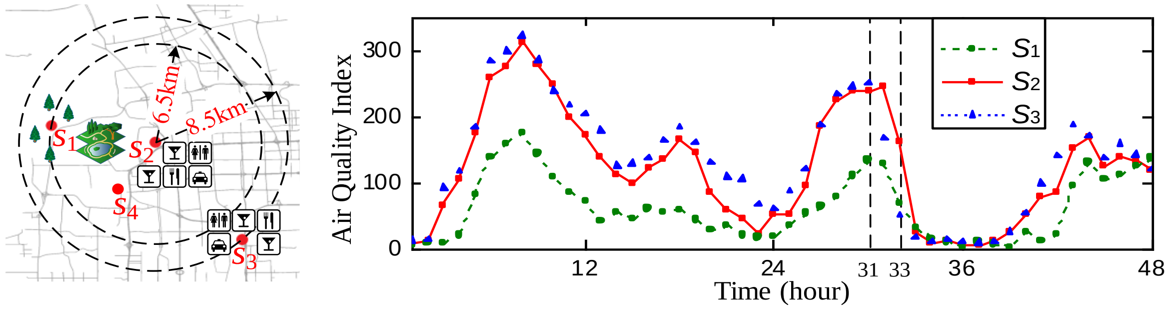
A) Missing situation

Filling Missing Values in Spatio-Temporal Data

- Difficulties
  - Random missing and block missing
    - ➢ Not handled by fixed learning models
  - Readings changing over time and location non-linearly
    - ➢ Not handled by simple interpolations



A) Missing situation

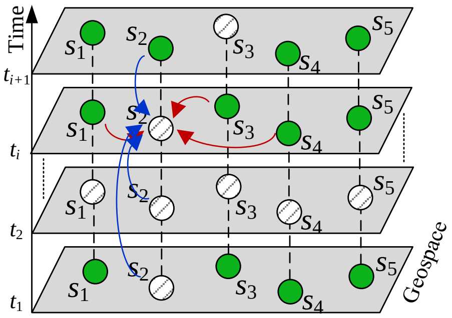


A) Geo-location of sensors

B) Air quality index over time

Fill Missing Values in Spatio-Temporal Datasets

- Achieve this goal from different perspectives
  - Spatial and Temporal perspectives
    - ➢ Spatial neighbors
    - ➢ Temporally adjacent time intervals
  - Global and temporal perspectives
    - ➢ Local: Recent context
    - ➢ Global: Long-term patterns



Temporal

Spatial

	t_1	t_2	……	t_{j-2}	t_{j-1}	t_j	t_{j+1}	t_{j+2}	……	t_{n-1}	t_n
s_1	230	230	….	205	164	185		188	….	223	249
s_2	200	188	….	173	136	X	146	185	….	199	255
s_3	118	93	….	72	56	59	44	78	….	99	111
⋮			⋮						⋮		
s_m	121	102	….	60	30	40	33	56	….	88	106

Global

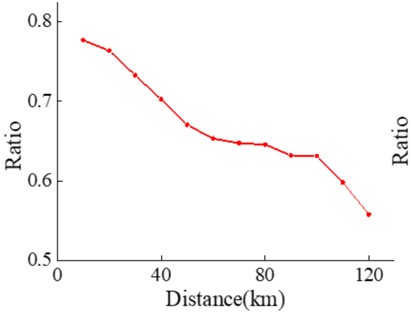
Global: Long-Term Knowledge

Spatial

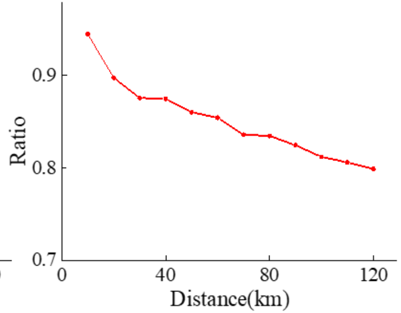
Inverse Distance Weighting (IDW)

$$\hat{v}_{gs} = \frac{\sum_{i=1}^{m} v_i * d_i^{-\alpha}}{\sum_{i=1}^{m} d_i^{-\alpha}}$$

Beijing from May 2014 to May 2015



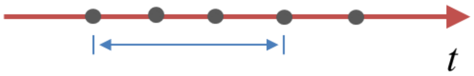
A) Air quality data



B) Humidity

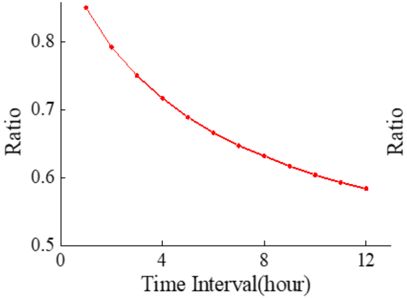
Temporal

Simple Exponential Smoothing (SES)

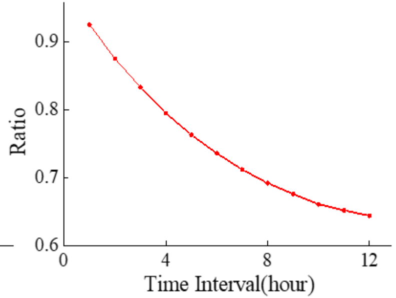


Time interval

$$\hat{v}_{gt} = \beta v_j + \beta(1-\beta) v_{j-1} + \cdots + \beta(1-\beta)^{t_j - 1} v_1$$



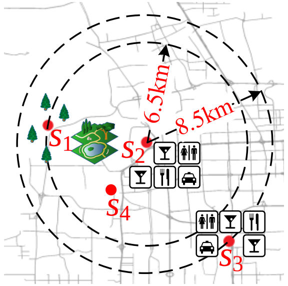
A) Air quality data



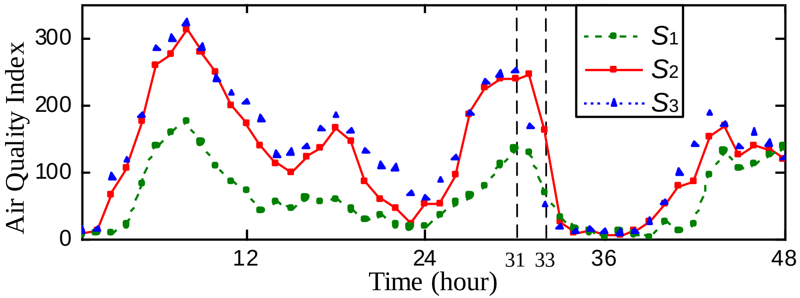
B) Humidity

Local: Recent Context

- Some situations break long-term patterns



A) Geo-location of sensors



B) Air quality index over time

Collaborative Filtering

- Sensors → users
- Time intervals → items

Temporal

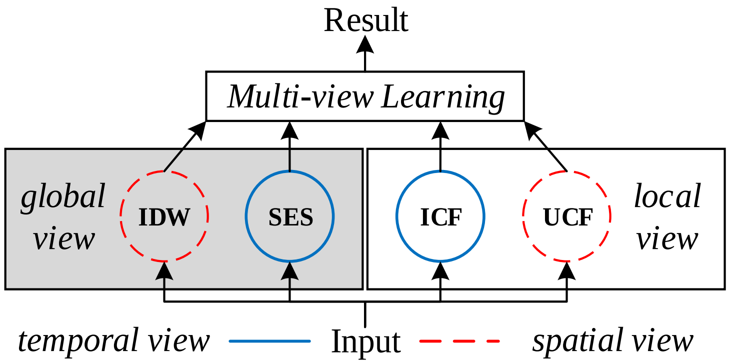
Spatial

	t_1	t_2	.....	t_{j-2}	t_{j-1}	t_j	t_{j+1}	t_{j+2}	.....	t_{n-1}	t_n
s_1	230	230	....	205	164	185		188	....	223	249
s_2	200	188	....	173	136		146	185	....	199	255
s_3	118	93	....	72	56	59	44	78	....	99	111
⋮			⋮					*Local*	⋮		
s_m	121	102	....	60	30	40	33	56	....	88	106

Global

Fill Missing Values in Spatio-Temporal Datasets

- A multi-view-based method
  - IDW: Inverse Distance Weighting
  - SES: Simple Exponential Smoothing
  - UCF: User-based Collaborative filtering
  - ICF: Item-based Collaborative filtering



$$\hat{v}_{mvl} = w_1 * \hat{v}_{gs} + w_2 * \hat{v}_{gt} + w_3 * \hat{v}_{ls} + w_4 * \hat{v}_{lt} + b$$

Temporal

Spatial

	t_1	t_2	……	t_{j-2}	t_{j-1}	t_j	t_{j+1}	t_{j+2}	……	t_{n-1}	t_n
s_1	230	230	….	205	164	185		188	….	223	249
s_2	200	188	….	173	136		146	185	….	199	255
s_3	118	93	….	72	56	59	44	78	….	99	111
⋮			⋮					*Local*	⋮		
s_m	121	102	….	60	30	40	33	56	….	88	106

Global

Experiments

		PM2.5	NO_2	Humidity	Wind Speed
Block missing	Spatial	2.2%	3.9%	9.8%	11.8%
	Temporal	3.5%	6.5%	9.6%	19.5%
General missing		8.2%	6.8%	4.6%	4.0%
Overall		13.3%	16.0%	21.5%	30.3%

Baselines

Method	Spatial	Temporal	Spatial + Temporal
Global	IDW	SES	IDW+SES
Lobal	UCF	ICF, ARMA	CF, NMF, stKNN
Global+Local	Kriging	SARIMA	AKE, DESM, NMF-MVL

Comparison among different methods (based on PM2.5)

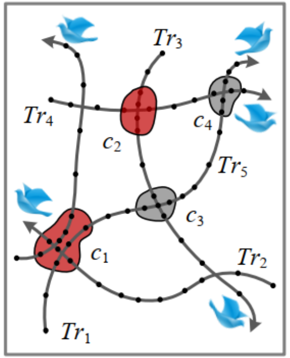
Method	General Missing		Spatial Block Missing		Temporal Block Missing		Sudden Change		Overall	
	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE
ARMA	22.61	0.331	29.26	0.369	\	\	51.11	0.567	27.47	0.394
Kriging	15.53	0.221	\	\	15.62	0.222	42.32	0.407	16.59	0.234
SARIMA	14.69	0.220	23.92	0.319	31.20	0.561	52.80	0.586	18.76	0.278
stKNN	12.84	0.188	19.91	0.235	12.72	0.226	35.13	0.390	14.00	0.201
DESM	13.65	0.191	19.24	0.233	12.66	0.224	42.87	0.425	15.59	0.228
AKE	13.34	0.195	19.08	0.229	12.14	0.22	41.54	0.403	14.27	0.211
IDW+SES	11.64	0.171	18.25	0.215	11.95	0.213	34.33	0.381	12.70	0.183
CF	12.20	0.178	19.27	0.234	12.25	0.218	34.91	0.388	13.40	0.193
NMF	11.21	0.163	18.98	0.239	12.73	0.217	34.37	0.381	13.08	0.188
NMF-MVL	11.16	0.162	18.97	0.238	12.66	0.217	34.33	0.380	13.06	0.187
ST-MVL	**10.81**	**0.158**	**17.85**	**0.217**	**11.71**	**0.208**	**33.15**	**0.368**	**12.12**	**0.174**

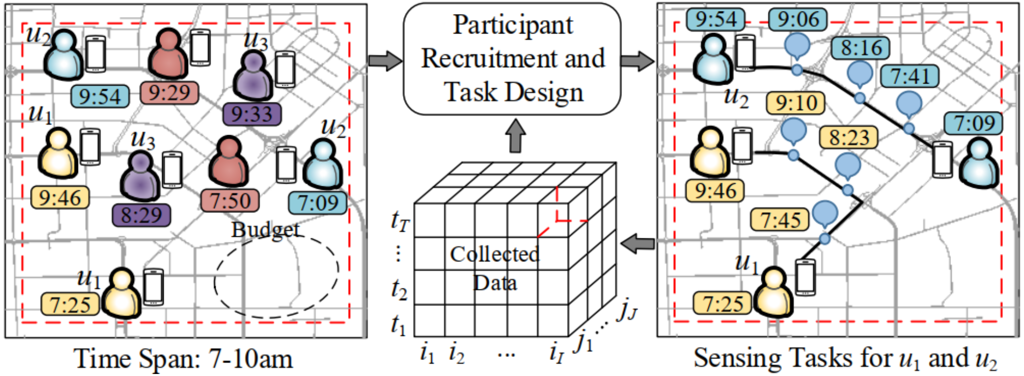
Challenges of Urban Sensing

Biased distribution

Data sparsity

Data missing

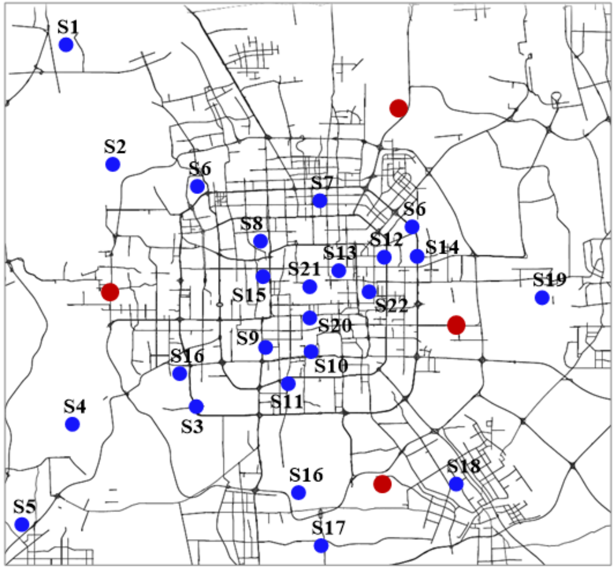


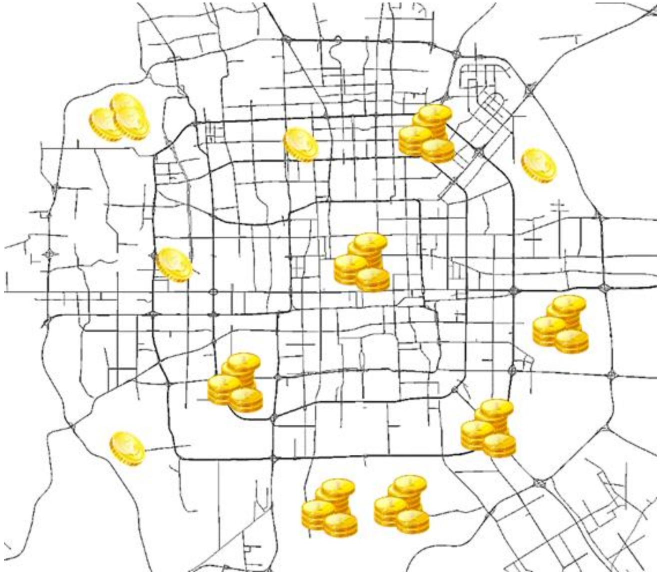


Resource deployment

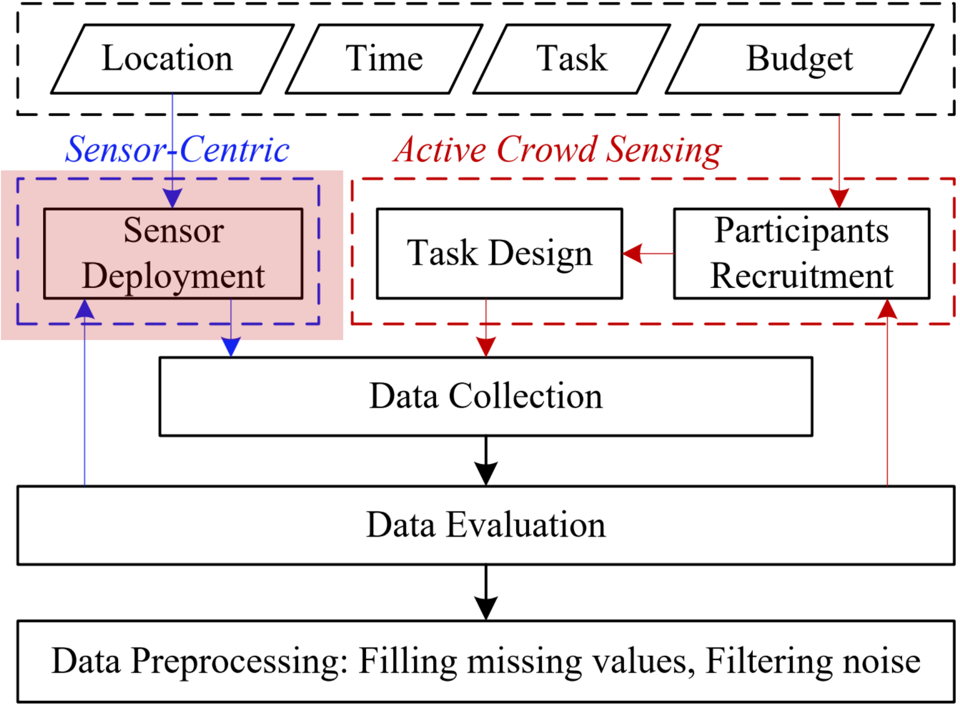
A limited resource (budget, labors, land…)

- Static sensing: Where to deploy sensor to maximize the gain?
- Crowdsensing: How to arrange the incentives dynamically?





General Framework of Urban Sensing



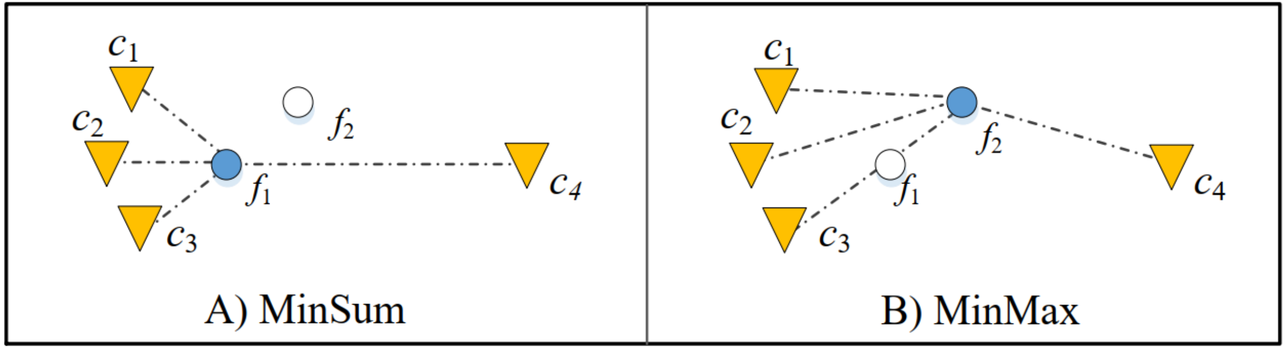
Resource Deployment and Location Selection

- Optimal meeting points problems
- Maximum coverage problem- submodular maximization
- Learning-to-rank problems
- Uncertainty minimization problems (entropy)

Optimal Meeting Points

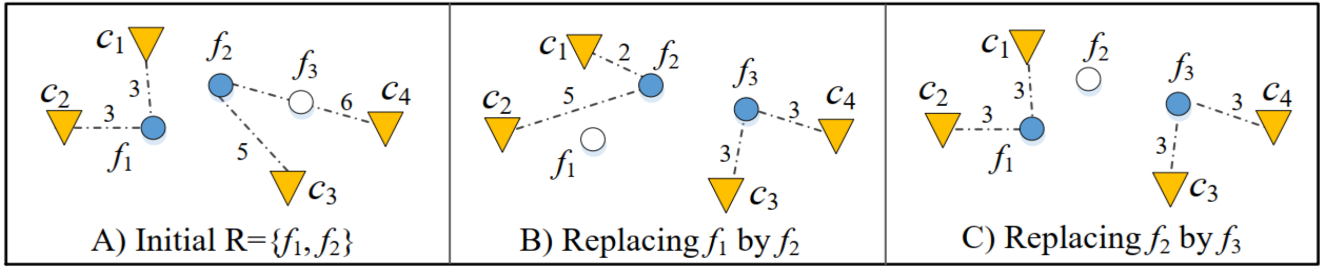
- locate k facilities such that
  - MinSum: Minimizing the average cost to reach all clients can be minimized
  - MinMax: Minimizing the maximum cost to reach those clients

NP-hard!

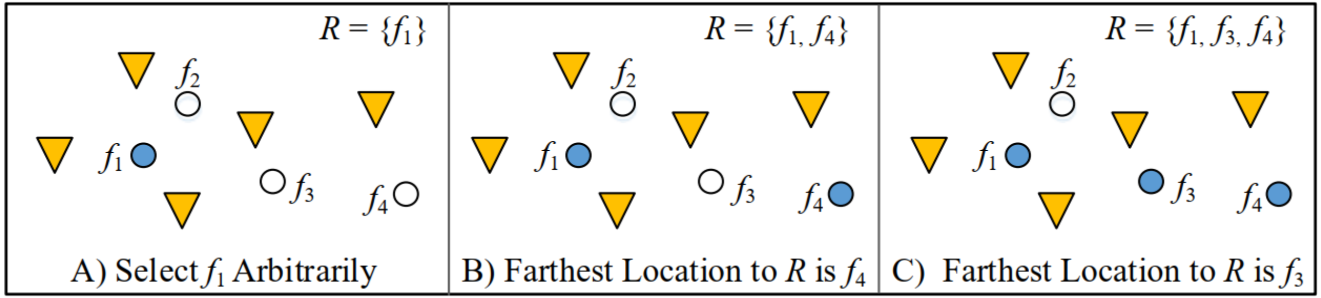


Optimal Meeting Points

- MinSum

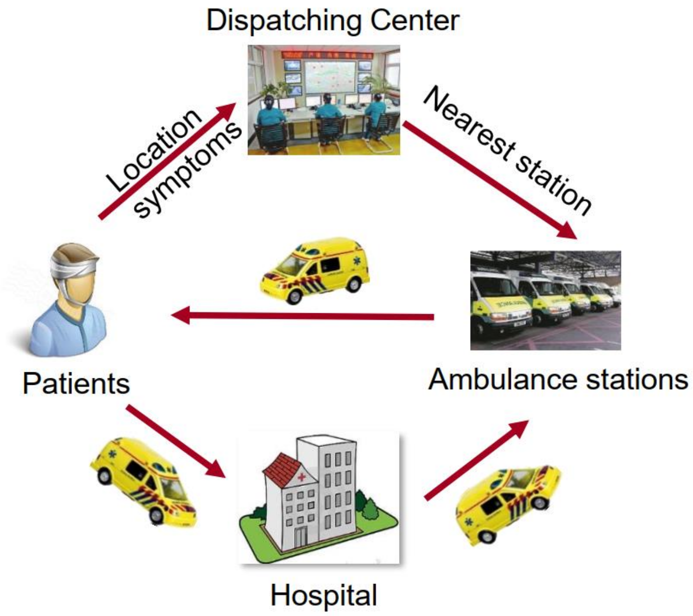


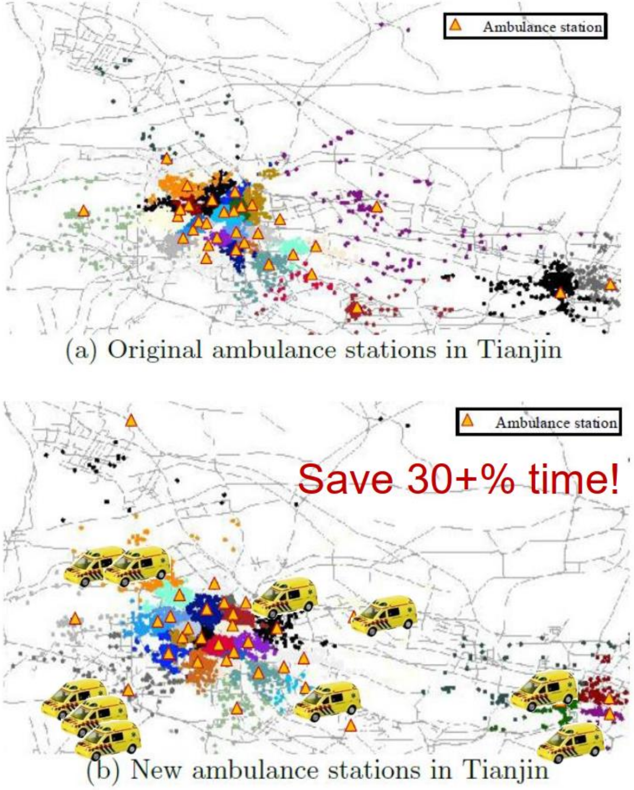
- MinMax



$d(f_3, R)$ denotes the minimum distance between f_3 and any facility in R

Improving Medical Emergency Services using Big Data



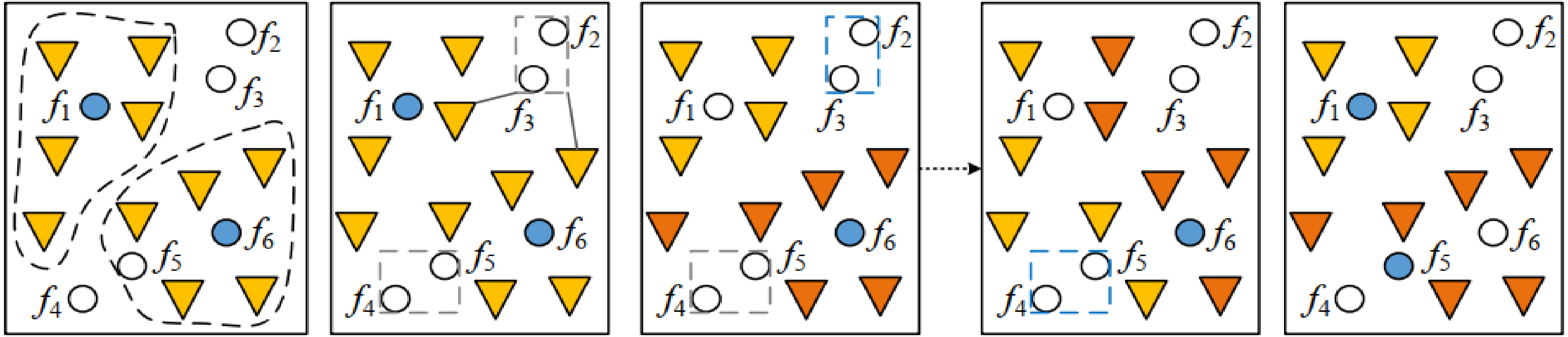


- Select locations for Ambulance Stations
- Dynamic ambulance allocation

Yuhong Li et al., Location Selection for Ambulance Stations: A Data-Driven Approach, ACM SIGSPATIAL 2015

Deploying Ambulance Stations Based on Optimal Meeting Point Model

- Methodology
  - k-medoids clustering
  - Candidate grouping
  - Iterations: pruning and replacement



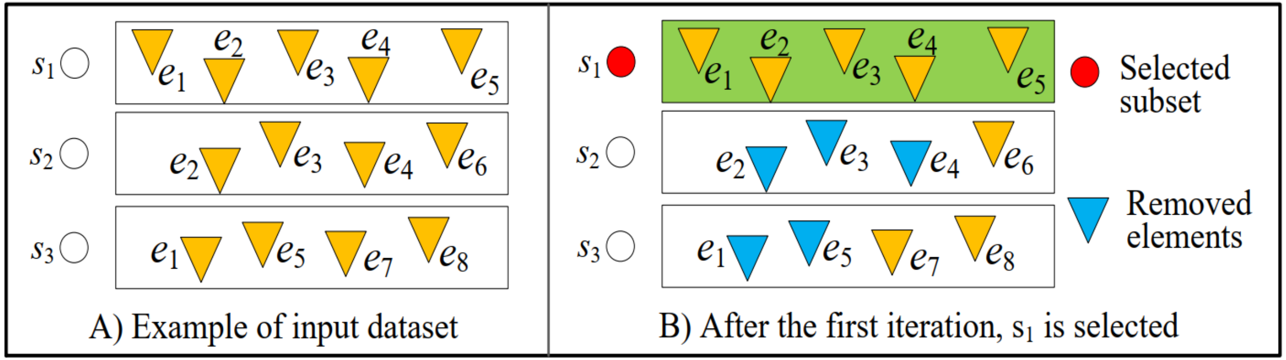
A) S1: Clustering B) S2: Grouping C) S3: Pruning and Replacement D) Final Results

Yuhong Li et al., [Location Selection for Ambulance Stations: A Data-Driven Approach](), ACM SIGSPATIAL 2015

Maximum Coverage Problem

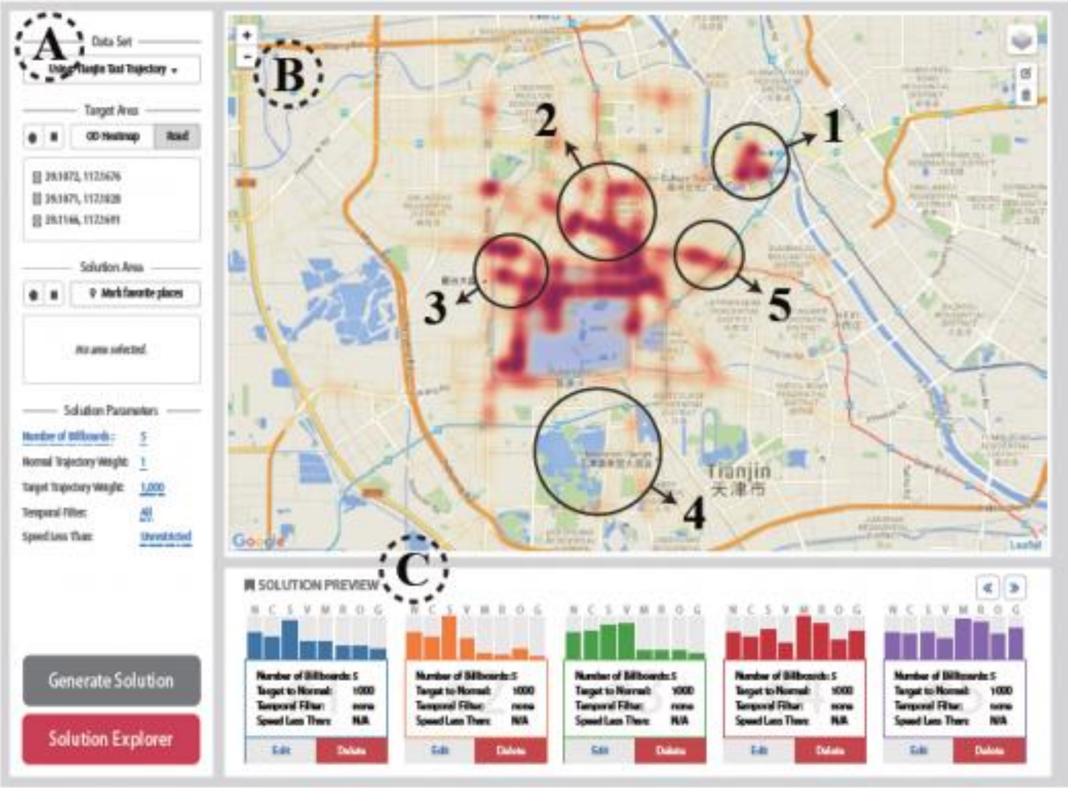
- Select a subset of (k) candidates to maximize
  - Nodes from a graph
  - Trajectories from a road network
  - Topic coverage from a document corpus

NP-hard!



$(1 - 1/e)$ approximation guarantee

Finding Top-k Most Influential Location Set



A submodular maximization problem, NP-hard!

Interactive Visual Data Analytics

Dongyu Liu et al., [SmartAdP: Visual Analytics of Large-scale Taxi Trajectories for Selecting Billboard Locations](), TVCG

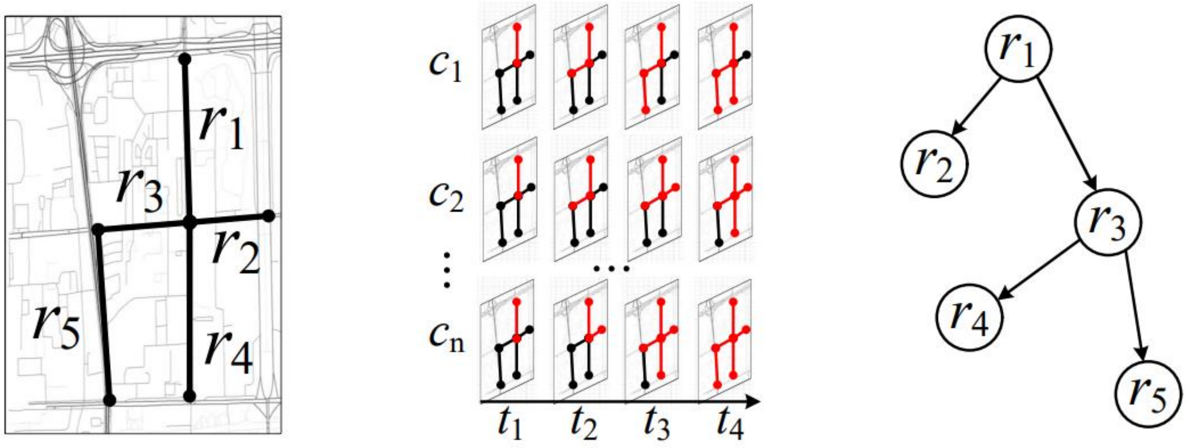
Inferring Traffic Cascading Patterns

A submodular maximization problem, NP-hard!

Yuxuan Liang et al., [Inferring Traffic Cascading Patterns](), SIGSPATIAL 2017

Inferring Traffic Cascading Patterns

- Target: to uncover how traffic congestion propagates thru road networks



(a) Road Network (b) Observed Cascades (c) Cascading Pattern

Yuxuan Liang et al., Inferring Traffic Cascading Patterns, SIGSPATIAL 2017

Inferring Traffic Cascading Patterns

- Also a maximum coverage problem

Cascading Pattern Construction

Model formulation

- Given a propagation tree T, the likelihood of a cascade c:
- Given a network G considering all possible trees T, the likelihood of a cascade c:

$$f(c|T) \Rightarrow f(c|G) = \sum_{T \in \mathrm{T}_c(G)} f(c|T)\underline{f(T|G)}$$

For a specific cascade, this part is equal.

simplify

$$f(c|G) \propto \sum_{T \in \mathrm{T}_c(G)} f(c|T)$$

Conditional independence

Joint likelihood of C

$$f(C|G) = \prod_{c \in C} f(c|G)$$

Optimization target: $\boxed{\hat{G}} = \arg\max_{|G| \le \boxed{k}} f(C|G)$ difficult to optimize

$\hat{G}$ best explains the observed cascades,
and the maximization is over all possible graphs G at most k edges

Alternative Optimization (CasInf)

$$\hat{G} = \arg\max_{|G| \le k} f(C|G)$$

1. matrix tree theorem
2. Combine environmental intensity
3. Log-likelihood

$$\hat{G} = \arg\max_{|G| \le k} F(C|G)$$

- Alternative target
- It satisfies submodularity, a natural diminishing returns property.

Optimization: greedy algorithm

Algorithm 1 Approximate algorithm for *CasInf*

Input:
k : the number of edges in cascading patterns we infer.
C : the set of cascades obtained in a time span.
$\mathcal{D}$: a constant denoting the spatial constraint.
Output: G: the inferred cascading pattern.

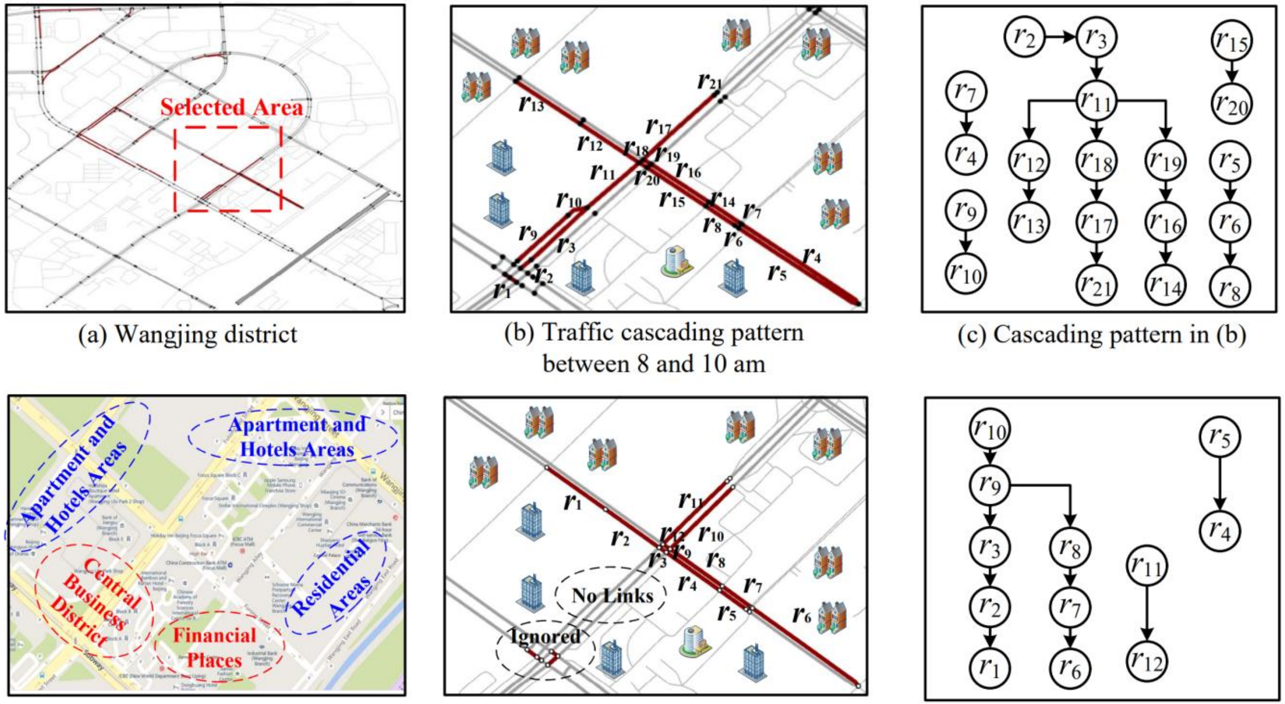
```
G ← K̃;
P ← all pairs (j, i): ∃c ∈ C with t_j < t_i and d_{j,i} < 𝒟
while |G| ≤ k do
   for all (j, i) ∈ P\G do
      δ_{j,i} = 0;
      for all c : t_j < t_i do
         ε_g ← environmental intensity inference
         w_c(m, n) ← weight of (m, n) in G ∪ (j, i);
         for all m : t_m < t_i and m ≠ j do
            δ_{c,j,i} = δ_{c,j,i} + w_c(m, i);
         end for
         δ_{j,i} = δ_{j,i} + log((δ_{c,j,i} + w_c(j,i)) / δ_{c,j,i});
      end for
   end for
   (j*, i*) ← argmax_{(j,i)∉G} δ_{j,i};
   G ← G ∪ (j*, i*);
end while
```

In each iteration, we select

$$e_i = \arg\max_{e \in G_i \backslash G_{i-1}} F(C|G_{i-1} \cup e) - F(C|G_{i-1}).$$

Yuxuan Liang et al., Inferring Traffic Cascading Patterns, SIGSPATIAL 2017

Inferring Traffic Cascading Patterns

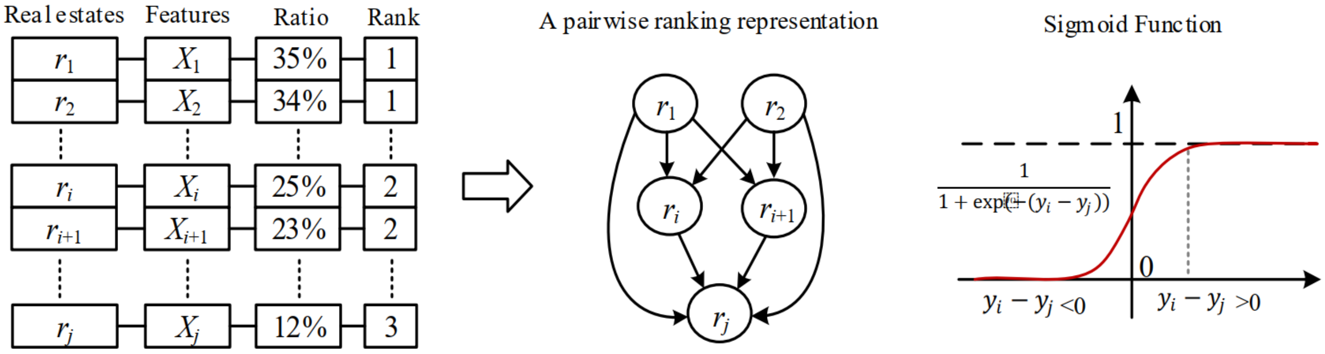


(a) Wangjing district
(b) Traffic cascading pattern between 8 and 10 am
(c) Cascading pattern in (b)
(d) Selected area
(e) Traffic cascading pattern between 7 and 9 pm
(f) Cascading pattern in (e)

Yuxuan Liang et al., [Inferring Traffic Cascading Patterns](), SIGSPATIAL 2017

Learning to Rank

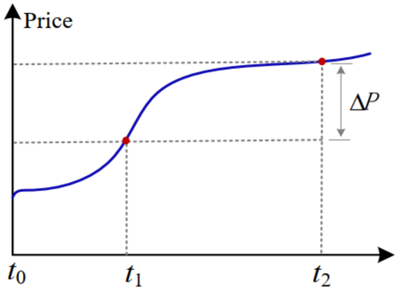
- Rank a set of locations so that the top-k best candidates can be selected to deploy resources or facilities
- Methods
  - Pointwise
  - Pairwise
  - Listwise



A) Ranking of Real Estates B) Graphical Representation of Orders C) Keeping Pairwise Orders

Ranking and Clustering Real Estates using Big Data

- Values (learned from big data)
  - Increase more in a rising market
  - Decrease less in a falling market



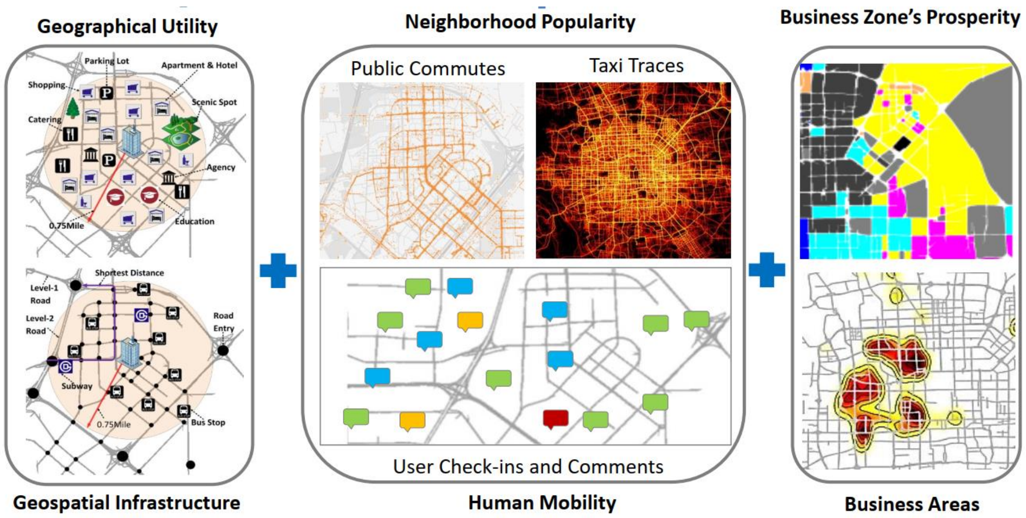
A) The price of a real estate

House	Increase	Rank↓
H1	35%	R1
H5	29%	R1
H4	13%	R2
….	….	….
H2	9%	R3
H3	2%	R3
H6	-1.5%	R4
H7	-6.1%	R5

B) Rank of estates by ΔP

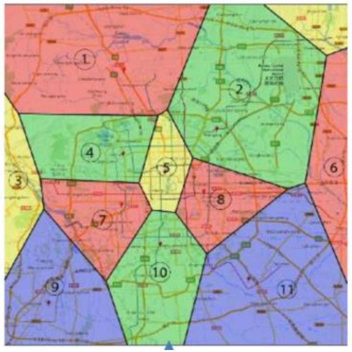
Yanjie Fu et al., [Sparse Real Estate Ranking with Online User Reviews and Offline Moving Behaviors](), ICDM 2014

Learning-to-Rank Problems



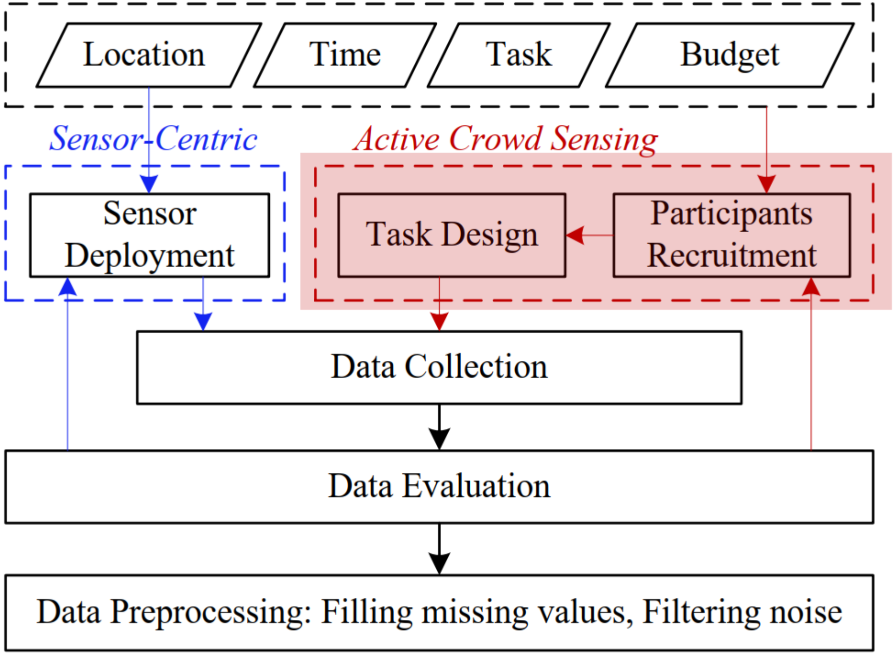
Ranking of Real Estates

Clusters of Real Estates



Yanjie Fu et al., Sparse Real Estate Ranking with Online User Reviews and Offline Moving Behaviors, ICDM 2014

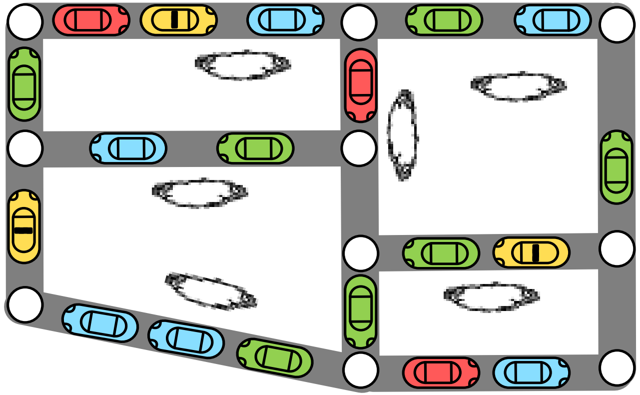
General Framework of Urban Sensing



Urban Sensing Based on Human Mobility

Ubicomp 2016

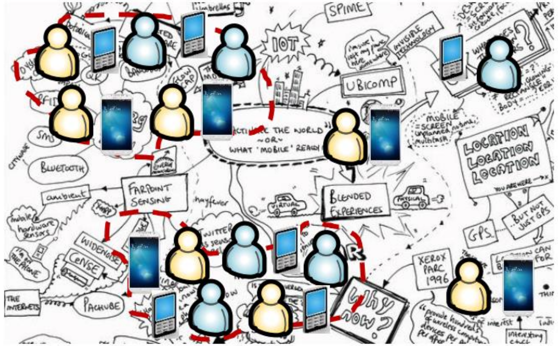
Shenggong Ji, et al. Urban Sensing Based on Human Mobility, UBICOMP 2014



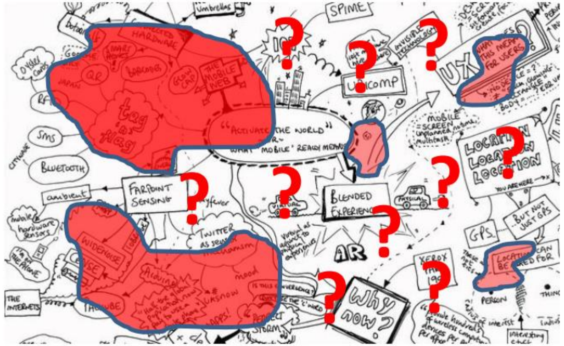
Urban Sensing

- Collecting urban data
  - Noise, temperature, air quality
  - Human as a sensor
- Brings challenges to
  - City-scale real-time monitoring
  - Further data analytics

Skewed human mobility

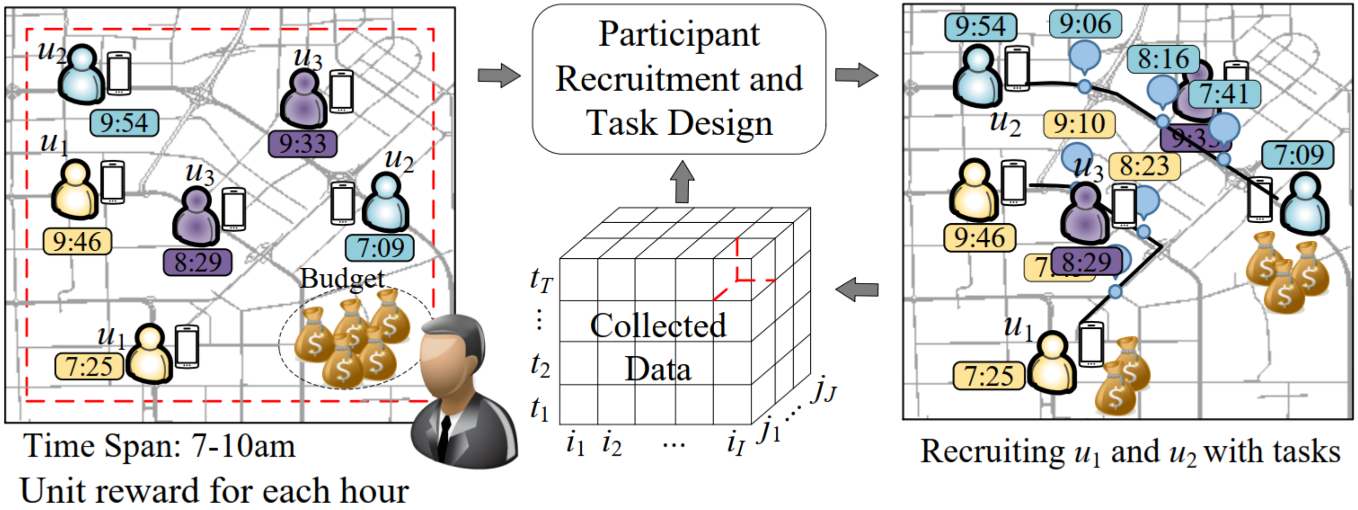


Imbalanced data coverage

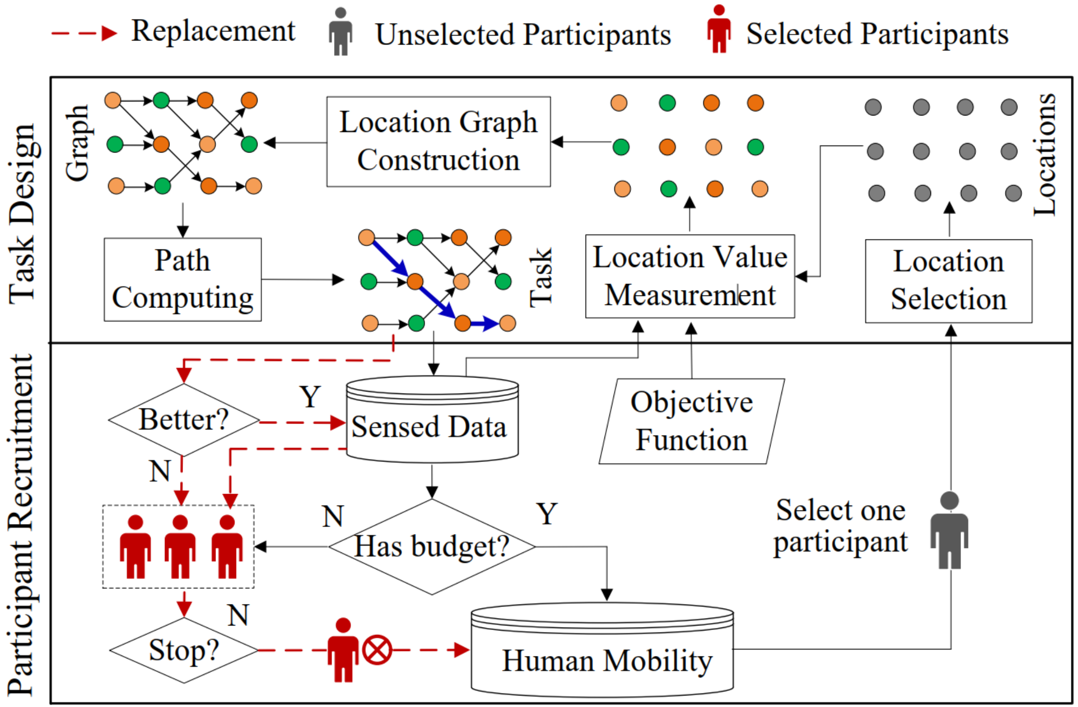


An Urban Sensing Framework

- Consider real-world human mobility
- Maximize the amount and balance of collected data
- Given a limited budget



Human Mobility-Based Urban Sensing Framework

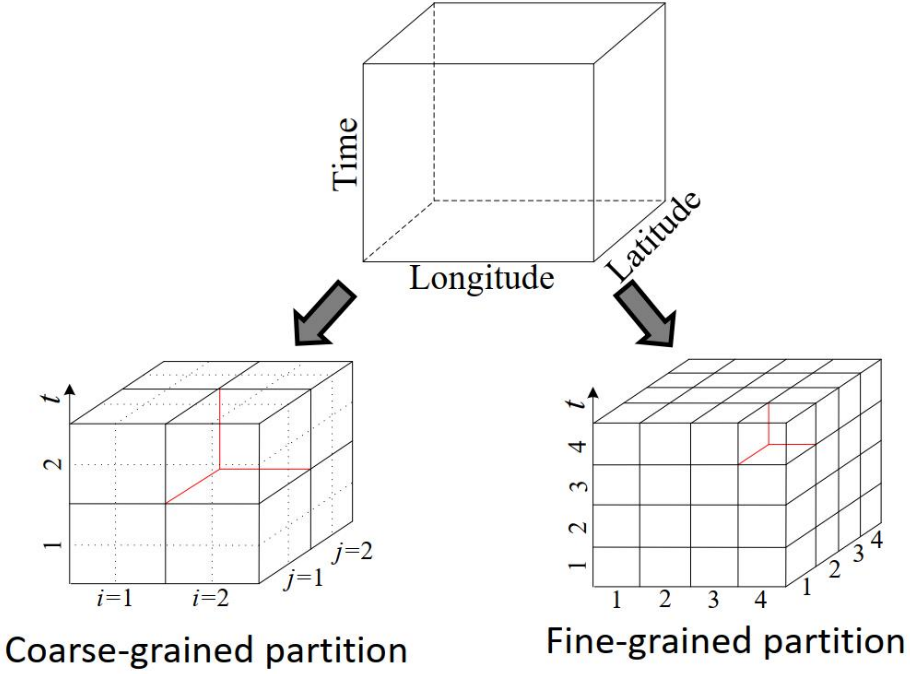


- A participant recruitment mechanism
  - random recruitment
  - replacement-based refinement

$$\max \phi = \alpha \times E + (1 - \alpha) \times \log_2 Q$$

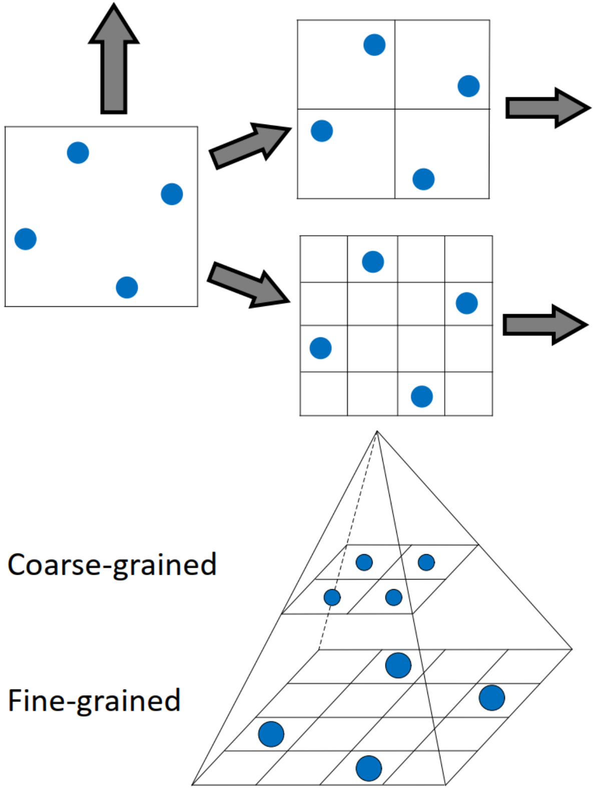
α: the relative preference of data balance to data amount

- application specific



Hierarchical Entropy-based Objective Function

Data amount: $Q = 4$



$$\left(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\right) \Rightarrow E_1 = 4 \times \left(-\frac{1}{4}\log_2\frac{1}{4}\right) = 2$$

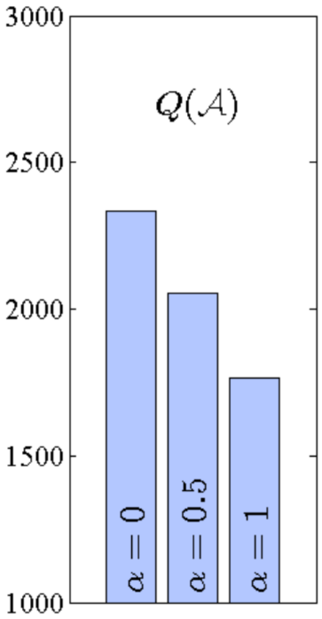
$$\left(0,\frac{1}{4},0,0,0,0,0,\frac{1}{4},\frac{1}{4},0,0,0,0,0,\frac{1}{4},0\right)$$

$$E_2 = 4 \times \left(-\frac{1}{4}\log_2\frac{1}{4}\right) + 12 \times (0\log_2 0) = 2$$

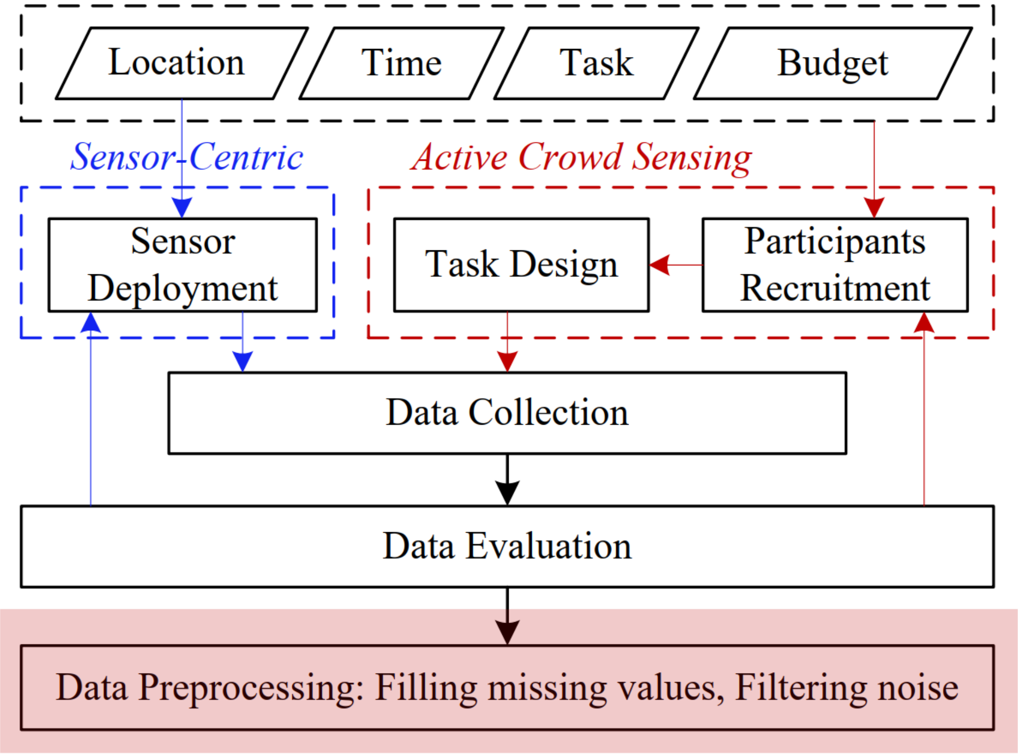
Data balance: $E = \dfrac{(2\times E_1 + 1\times E_2)}{2} = 3$

Real-World Evaluation

- Datasets
  - Human mobility dataset from a real-world noise sensing experiment
    - ➢ Sensing region: 6.6km × 3.3km
    - ➢ Sensing time interval: 6:00 am ~ 22:00 pm
  - 244 participant candidates with mobility information



General Framework of Urban Sensing



Q&A

Thanks!

CityMind Lab

amazon Microsoft Tencent Alibaba.com 京东 JD.COM CAINIAO 菜鸟