23 March 2017

CISS 2017, Baltimore, MD

Distributed Learning of Human Mobility Patterns From Cellular Network Data

Tong Wu[1], Raif Rustamov[2] and *Colin Goodall[3]*

[1]Department of Electrical and Computer Engineering

Rutgers, The State University of New Jersey

[2]AT&T Labs – Research, Bedminster NJ

[3]AT&T Labs – Research, Middletown NJ

An overview of dictionary learning

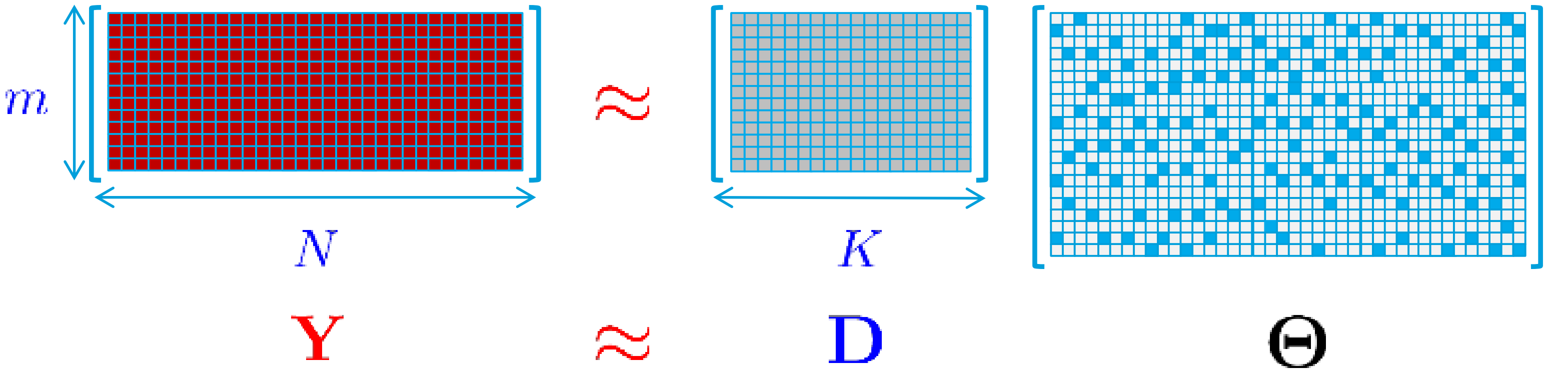
Dictionary learning

A data-driven approach to learn from $\mathbf{Y}$ a sparsifying, overcomplete, approximating basis $\boldsymbol{\Theta}$

N feature vectors, *K* atoms = basis elements in $\mathbf{D}$, each with dimension *m* features

$\boldsymbol{\Theta}$ is sparse, at most *s* non-zero elements in each column

$$N \gg K > m \;\; (\gg s)$$



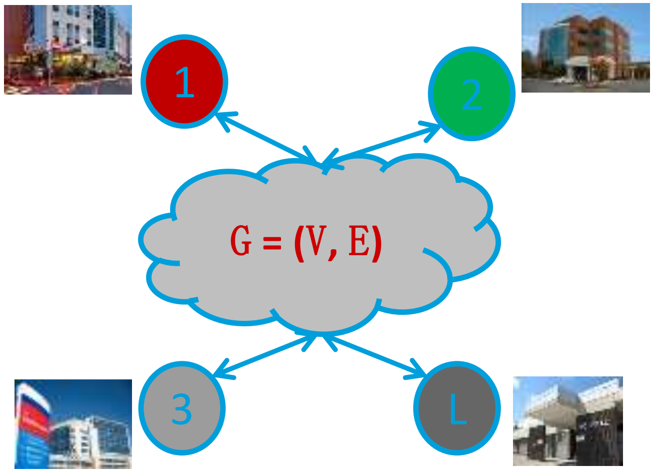
Applications

- Image de-noising, classification, in-painting, etc [Elad et al. 2006, Mairal et al. 2008]
- Action recognition [Guha et al. 2012, Qiu et al. 2013]
- Novel document detection [Kasiviswanathan et al. 2012]
- Human mobility understanding [Wu et al. 2017]

Distributed learning of mobility patterns

Problem: Learning traffic patterns/prevalent commute routes to support Urban Mobility Modeling and Smart Cities

Call Detail Records (CDRs) Data, includes IMSI (a unique number associated with the cell phone), time stamp, latitude, and longitude of cell site locations.



Challenge: How to learn human mobility patterns when CDRs are distributed across a set of interconnected data centers?

Constraints, *for the purposes of this talk*:-

- Local CDRs are “big” and “raw data” cannot be shared with other sites

  In practice, latency and use of network capacity

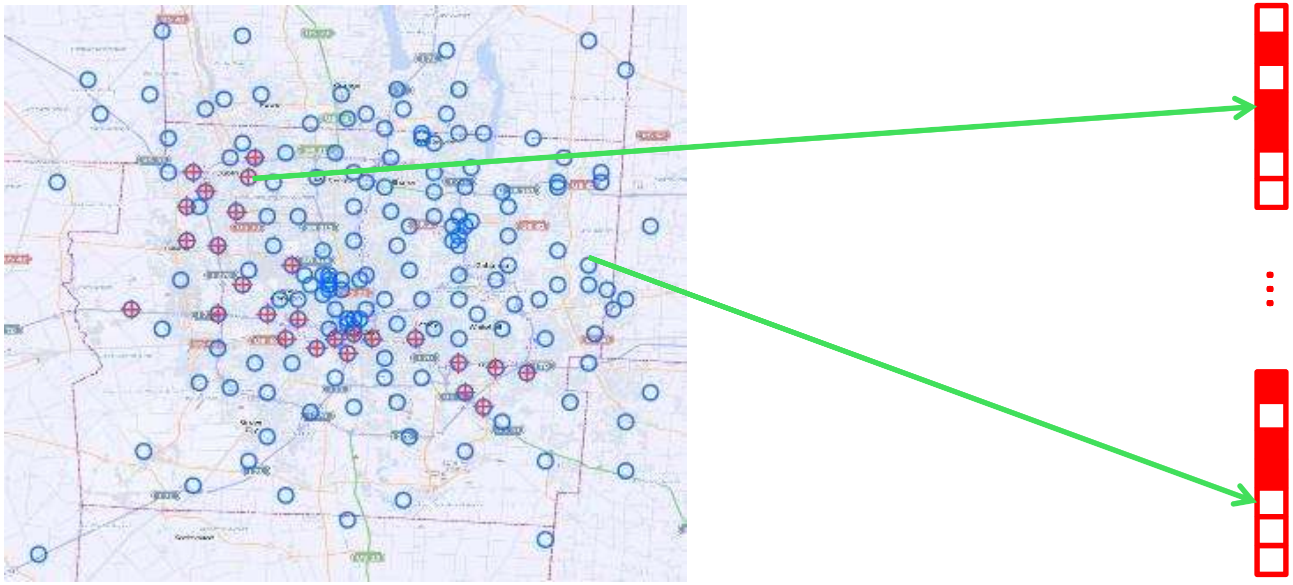
- The graph of interconnections cannot be described by a complete graph
- Sites do not have knowledge of global network topology

  Re dynamic and sensor network applications (?)

RUTGERS AT&T

Extract features from CDRs for dictionary learning

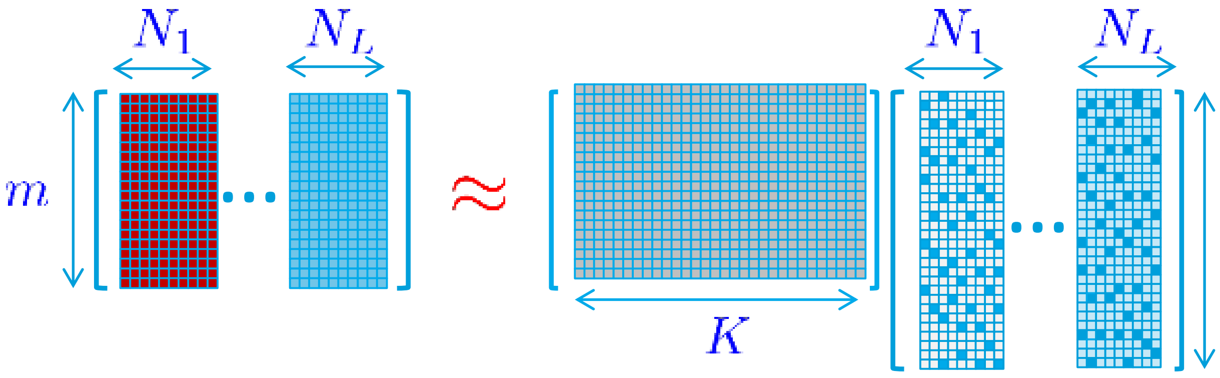
- Slippy tile encoding of lat/long; each slippy tile is approx. 600m x 600m; Open Street Maps
- Columbus OH region contains 175 slippy tiles.
- For every user whose CDRs are collected in one data center/node, create a binary feature vector (also defined as a path) indicating the incidence between this user and the slippy tiles in the course of one day. *For the purposes of this talk, the features are unordered*.



$\mathbf{y}_i \in \mathbb{R}^{175}$

Basic idea: each path can be described as a combination of some typical traffic routes (aka atoms) taken by drivers in Columbus. Take s = 3 or 4 atoms for each path.

Distributed data: Mathematical problem formulation



$$\mathbf{Y} \quad \approx \quad \mathbf{D} \quad \boldsymbol{\Theta}$$

$$\mathbf{Y} = \left[\mathbf{Y}_1 \quad \mathbf{Y}_2 \quad \cdots \quad \mathbf{Y}_L\right], \mathbf{Y}_\ell \in \mathbb{R}^{m \times N_\ell}$$

Dictionary learning in the centralized setting

Learn an overcomplete $m \times K$ dictionary $\boldsymbol{D}$ such that the training samples $\boldsymbol{Y}_\ell$ at each site can be approximated by linear combinations of a small number of columns in $\boldsymbol{D}$.

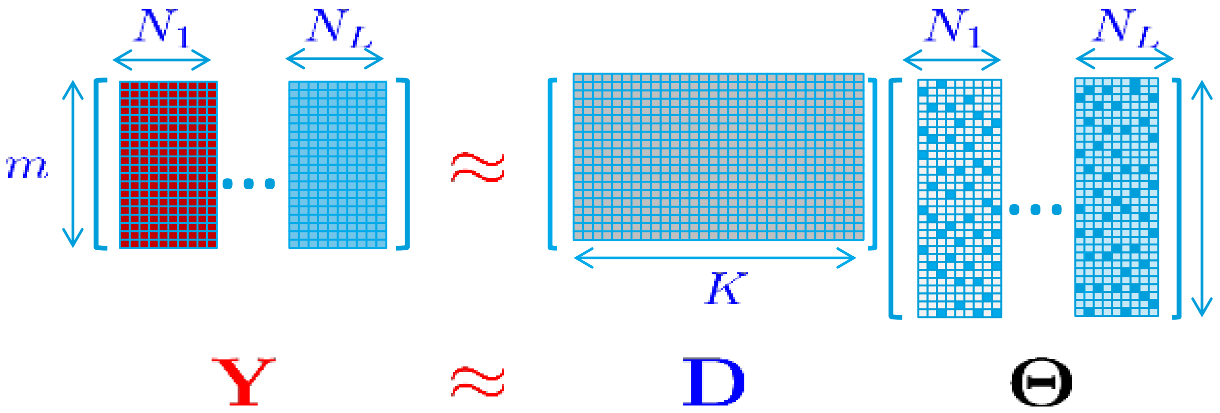
$$(\mathbf{D}^*, \boldsymbol{\Theta}^*) = \arg\min_{\mathbf{D}, \boldsymbol{\Theta} \geq 0} \|\mathbf{Y} - \mathbf{D}\boldsymbol{\Theta}\|_F^2 \text{ s.t. } \forall i, \ \|\boldsymbol{\theta}_i\|_0 \leq s, \ \forall k, \ \|\mathbf{d}_k\|_2 = 1$$

distributed across L sites

Collaborative dictionary learning

Collaboratively learn an overcomplete dictionary $\boldsymbol{D}_\ell^*$ at each site ℓ such that $\boldsymbol{D}_\ell^* \approx \boldsymbol{D}^*$.

Family of K-SVD algorithms



$$\mathbf{Y} = \left[\mathbf{Y}_1 \quad \mathbf{Y}_2 \quad \cdots \quad \mathbf{Y}_L\right], \mathbf{Y}_\ell \in \mathbb{R}^{m \times N_\ell}$$

Centralized dictionary learning, K-SVD

dictionary D

Collaborative dictionary learning, Cloud K-SVD (same picture, repeated L times)

Collaboratively learn an overcomplete dictionary $\boldsymbol{D}_\ell^*$ at each site ℓ such that $\boldsymbol{D}_\ell^* \approx \boldsymbol{D}^*$.

Non-negative dictionary learning, NN-K-SVD

$\mathbf{\Theta}$ non-negative

centralized NN-K-SVD and collaborative / Cloud NN-K-SVD

Local dictionary learning

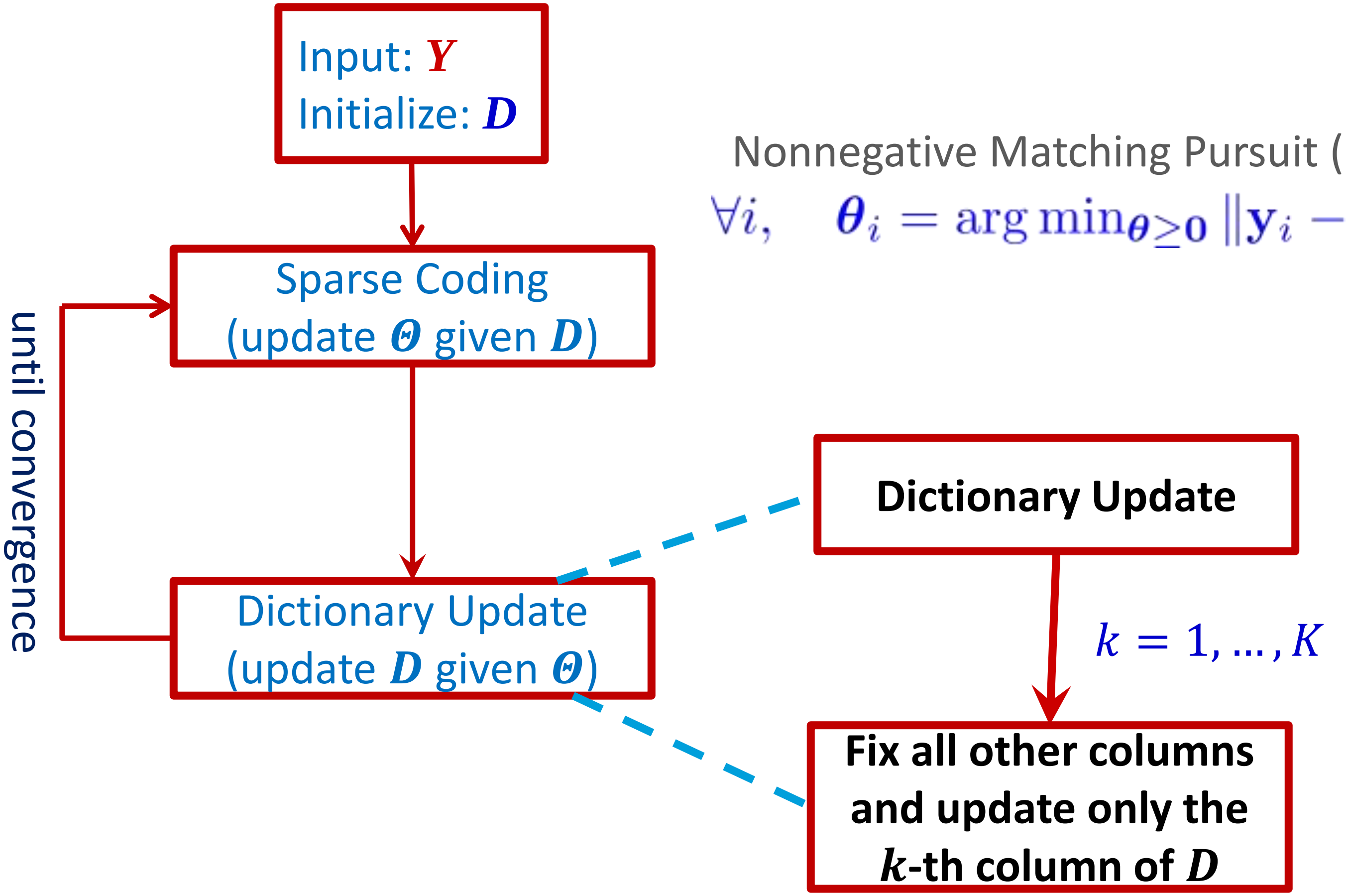
Learn an overcomplete dictionary $\boldsymbol{D}_\ell^*$ at each site ℓ without communication between sites.

A brief review of Nonnegative K-SVD

$$(\mathbf{D}^*, \mathbf{\Theta}^*) = \arg\min_{\mathbf{D}, \mathbf{\Theta} \geq \mathbf{0}} \|\mathbf{Y} - \mathbf{D}\mathbf{\Theta}\|_F^2 \text{ s.t. } \forall i, \ \|\boldsymbol{\theta}_i\|_0 \leq s, \ \forall k, \ \|\mathbf{d}_k\|_2 = 1$$

Nonnegative Matching Pursuit (NMP) [Peharz et al. 2010]

$$\forall i, \quad \boldsymbol{\theta}_i = \arg\min_{\boldsymbol{\theta} \geq \mathbf{0}} \|\mathbf{y}_i - \mathbf{D}\boldsymbol{\theta}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\theta}\|_0 \leq s$$



Input: $\boldsymbol{Y}$

Initialize: $\boldsymbol{D}$

Sparse Coding (update $\boldsymbol{\Theta}$ given $\boldsymbol{D}$)

Dictionary Update (update $\boldsymbol{D}$ given $\boldsymbol{\Theta}$)

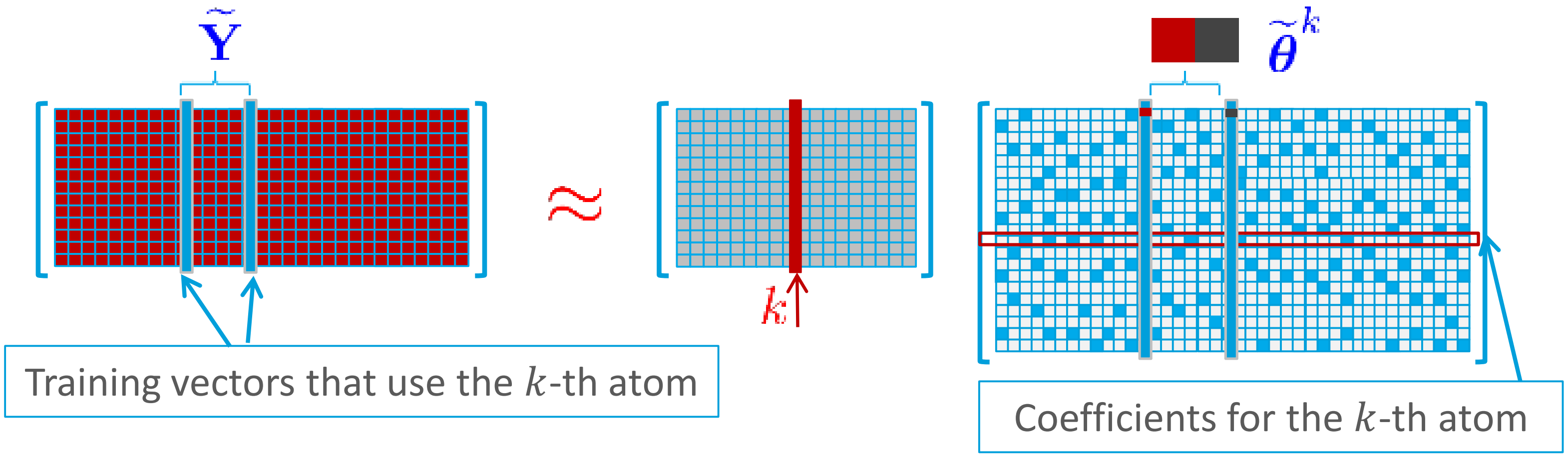
until convergence

Dictionary Update

$k = 1, \ldots, K$

Fix all other columns and update only the $\boldsymbol{k}$-th column of $\boldsymbol{D}$

Dictionary update in Nonnegative K-SVD



Training vectors that use the k-th atom

Coefficients for the k-th atom

Basic idea

Update the k-th atom of the dictionary by focusing only on the training data utilizing it

$$(\mathbf{d}_k, \widetilde{\boldsymbol{\theta}}^k) = \arg\min_{\mathbf{d},\mathbf{g}\geq\mathbf{0}} \| \underbrace{(\widetilde{\mathbf{Y}} - \sum_{j\neq k} \mathbf{d}_j \widetilde{\boldsymbol{\theta}}^j)}_{\mathbf{E}_k} - \mathbf{dg}\|_F^2 \quad \text{s.t. } \|\mathbf{d}\|_2^2 = 1$$

Solution

Initialize $\boldsymbol{d} = \boldsymbol{a}$ and $\boldsymbol{g} = \sigma\boldsymbol{b}$ and set the negative entries of $\boldsymbol{d}$ and $\boldsymbol{g}$ to be zero, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are the dominant left and right singular vectors of $\boldsymbol{E}_k$

Repeat J times

$$\boldsymbol{d} = \frac{\boldsymbol{E}_k \boldsymbol{g}^T}{\boldsymbol{g}\boldsymbol{g}^T}, \boldsymbol{d} = \boldsymbol{d} \otimes [\boldsymbol{d} > \boldsymbol{0}] \qquad \boldsymbol{g} = \frac{\boldsymbol{d}^T \boldsymbol{E}_k}{\boldsymbol{d}^T \boldsymbol{d}}, \boldsymbol{g} = \boldsymbol{g} \otimes [\boldsymbol{g} > \boldsymbol{0}]$$

Collaborative dictionary learning - cloud NN-K-SVD

$$(\mathbf{D}^*, \mathbf{\Theta}^*) = \arg\min_{\mathbf{D},\mathbf{\Theta}\geq 0} \|\mathbf{Y} - \mathbf{D}\mathbf{\Theta}\|_F^2 \text{ s.t. } \forall i,\ \|\boldsymbol{\theta}_i\|_0 \leq s,\ \forall k,\ \|\mathbf{d}_k\|_2 = 1$$

Distributed across L sites: $\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & \mathbf{Y}_2 & \cdots & \mathbf{Y}_L \end{bmatrix}, \mathbf{Y}_\ell \in \mathbb{R}^{m \times N_\ell}$

How to go from NN-K-SVD to cloud NN-K-SVD?

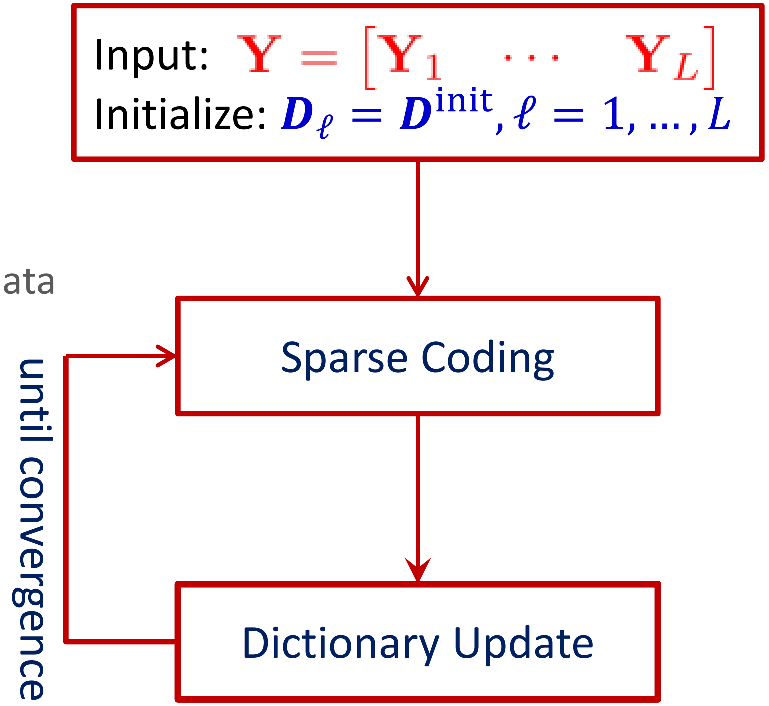
Part 1: How to distribute sparse coding?

Do not need to distribute. Instead:
Each site ℓ locally computes sparse codes for its local training data

$$\forall \ell = 1, \ldots, L, \forall i = 1, \ldots, N_\ell$$
$$\boldsymbol{\theta}_{\ell,i} = \arg\min_{\boldsymbol{\theta}\geq 0} \|\mathbf{y}_{\ell,i} - \mathbf{D}_\ell \boldsymbol{\theta}\|_2^2 \text{ s.t. } \|\boldsymbol{\theta}\|_0 \leq s$$

Part 2: How to distribute dictionary update?

Additional methodology needed, next slide



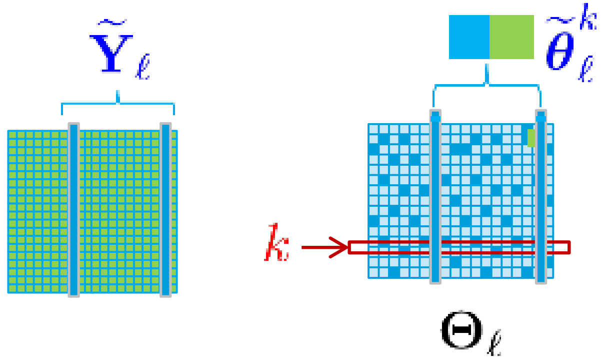
Input: $\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{Y}_L \end{bmatrix}$
Initialize: $\boldsymbol{D}_\ell = \boldsymbol{D}^{\text{init}}, \ell = 1, \ldots, L$

Sparse Coding

Dictionary Update

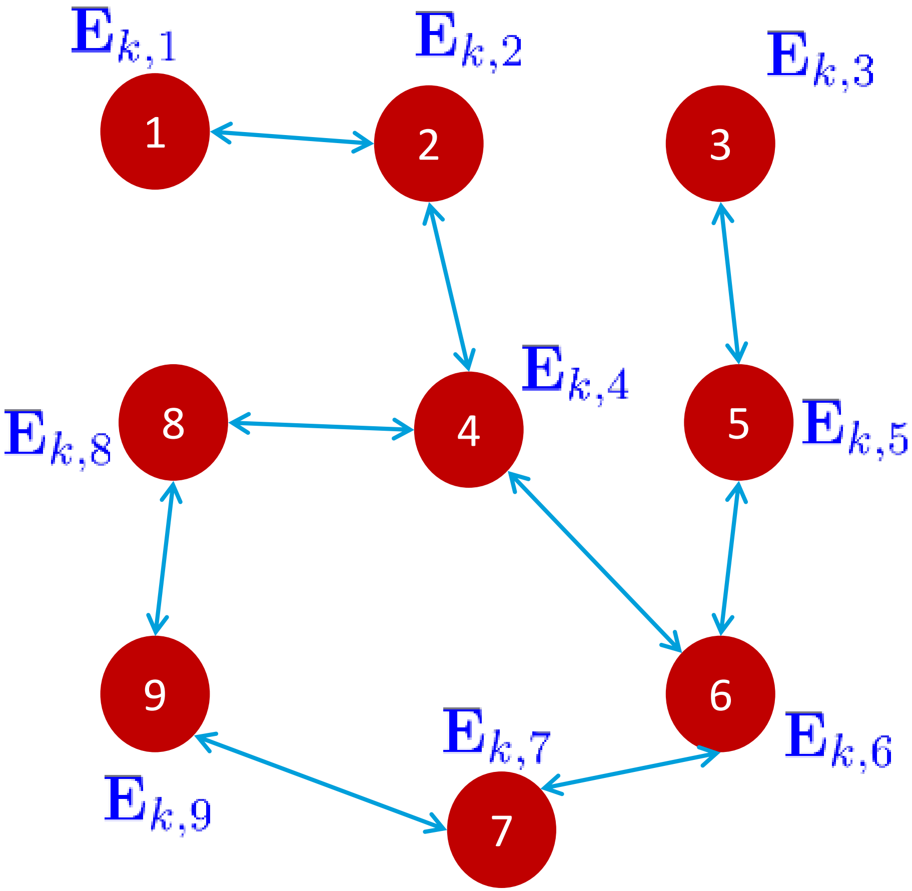
until convergence

Collaborative dictionary update in cloud NN-K-SVD

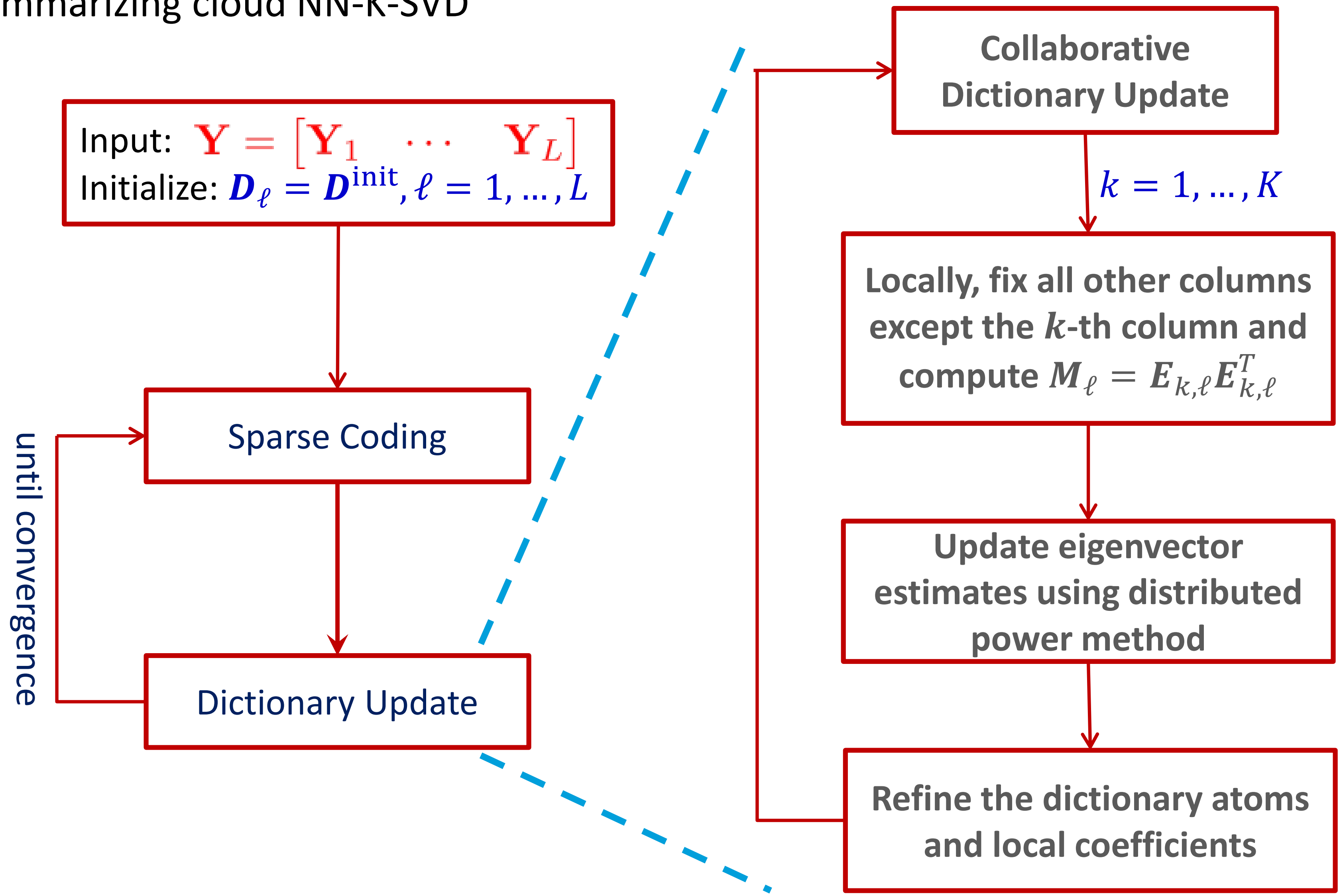

$\widetilde{\mathbf{Y}}_\ell$
$\widetilde{\boldsymbol{\theta}}_\ell^k$
k
$\boldsymbol{\Theta}_\ell$

$$\mathbf{E}_{k,\ell} = \widetilde{\mathbf{Y}}_\ell - \sum_{j \neq k} \mathbf{d}_{\ell,j} \widetilde{\boldsymbol{\theta}}_\ell^j$$

$$\mathbf{E}_k = \begin{bmatrix} \mathbf{E}_{k,1} & \mathbf{E}_{k,2} & \cdots & \mathbf{E}_{k,L} \end{bmatrix}$$



Summarizing cloud NN-K-SVD



Input: $\mathbf{Y} = [\mathbf{Y}_1 \quad \cdots \quad \mathbf{Y}_L]$

Initialize: $\boldsymbol{D}_\ell = \boldsymbol{D}^{\text{init}}, \ell = 1, \dots, L$

Sparse Coding

Dictionary Update

until convergence

Collaborative Dictionary Update

$k = 1, \dots, K$

Locally, fix all other columns except the k-th column and compute $\boldsymbol{M}_\ell = \boldsymbol{E}_{k,\ell}\boldsymbol{E}_{k,\ell}^T$

Update eigenvector estimates using distributed power method

Refine the dictionary atoms and local coefficients

Experimental Setup

- Three days of data: May 7(Sat), May 11(Wed), and May 12(Thur).
- Divide the data for each day into 12,000 devices (*training*) and 4,000 devices (*test*).
- Generate a random network with 10 nodes for distributed learning
- Each node has 1,200 training samples.
- Parameters: $m = 175$, $K = 200$, $s = 4$ for weekdays and $s = 3$ for weekends.

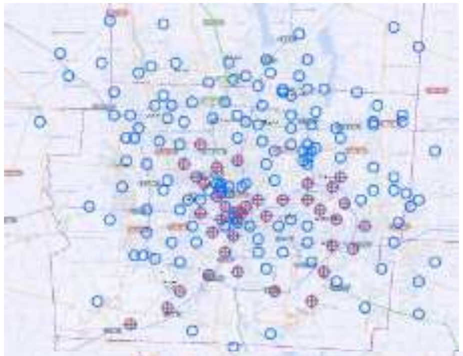
Comparison metric:

For every test data $\boldsymbol{y}$, compute its sparse representation coefficients $\boldsymbol{\theta}$ with respect to the learned dictionary $\boldsymbol{D}$ ($\boldsymbol{D}_\ell$'s in the distributed setting). Representation error

$$\epsilon(\boldsymbol{y}) = \|\boldsymbol{y} - \boldsymbol{D\theta}\|_2^2 / \|\boldsymbol{y}\|_2^2.$$

	centralized K-SVD	local K-SVD	cloud K-SVD	centralized NN-K-SVD	local NN-K-SVD	cloud NN-K-SVD
May 7	0.240	0.301	0.283	0.237	0.274	0.238
May 11	0.252	0.325	0.277	0.246	0.279	0.246
May 12	0.252	0.326	0.278	0.247	0.280	0.247

Path/Route visualization

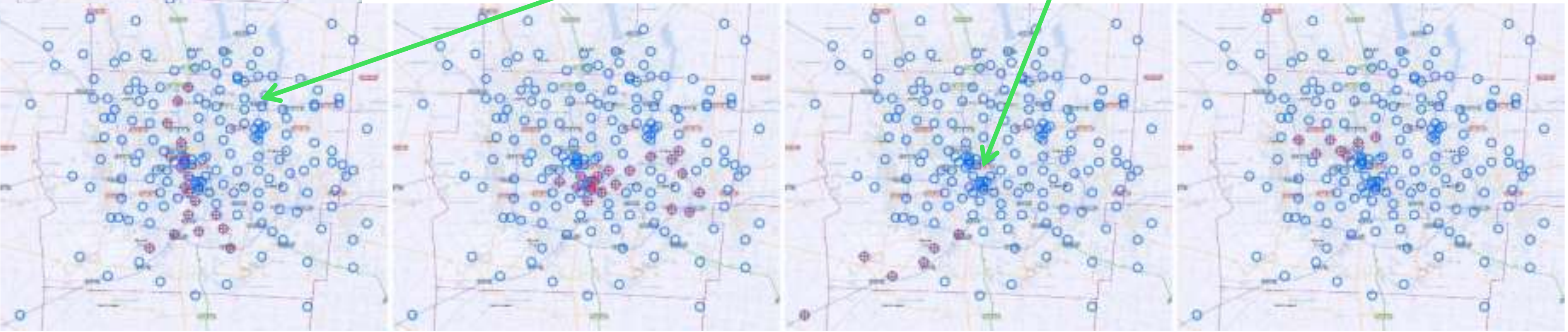


test path

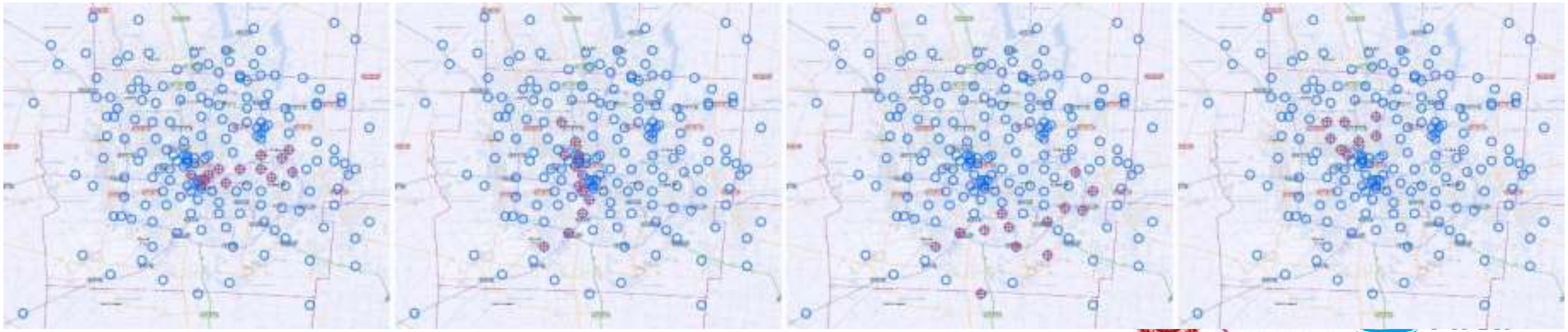
May 12

Overlapping!

associated atoms for K-SVD



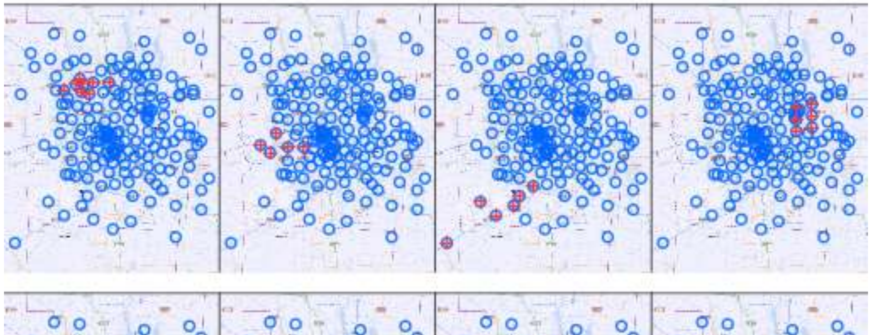
associated atoms for NN-K-SVD

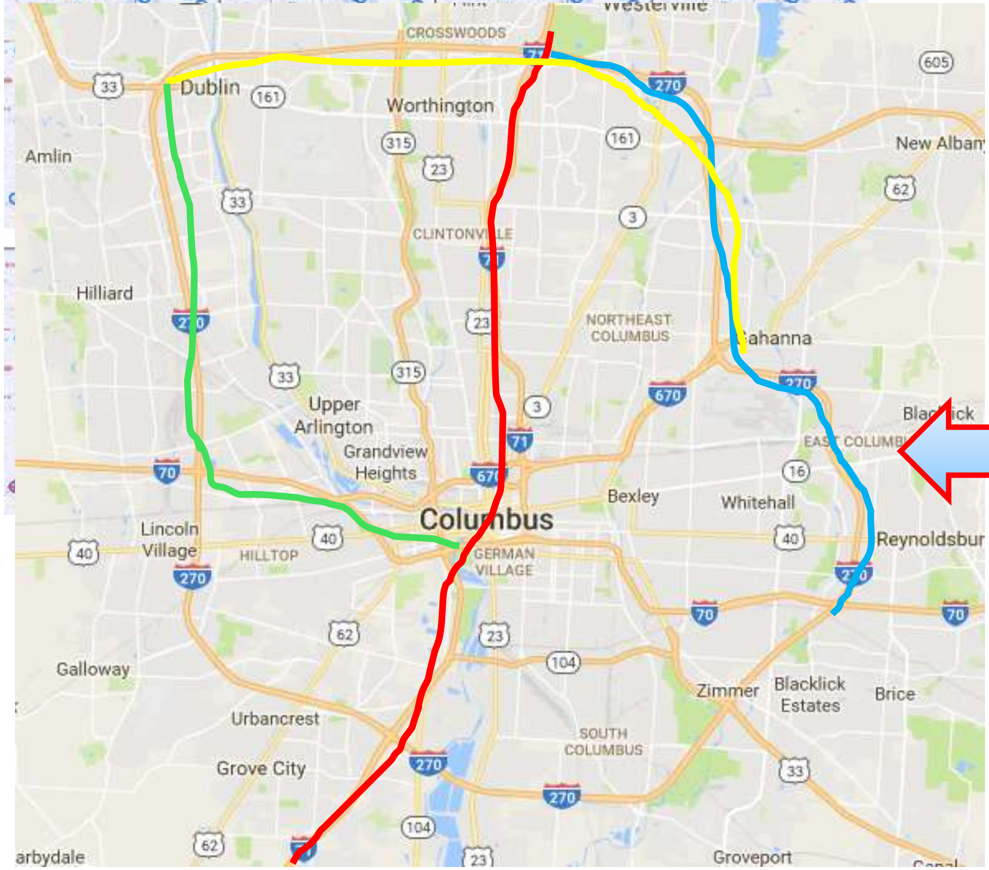


Frequent routes of test data - Atoms

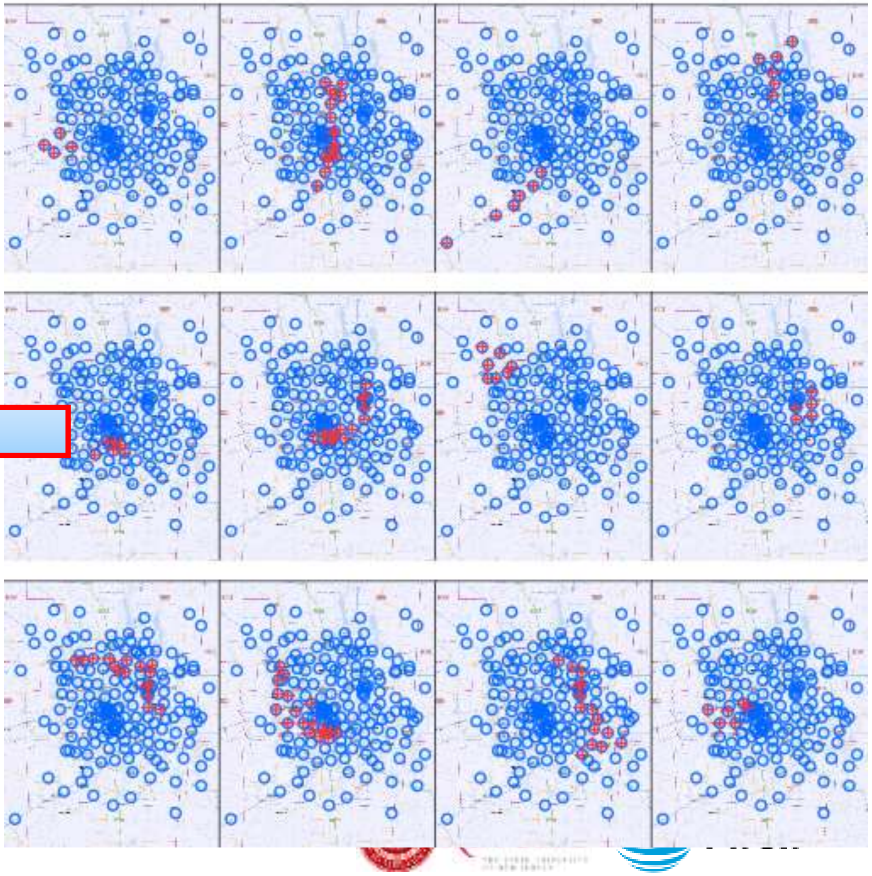
May 12

K-SVD

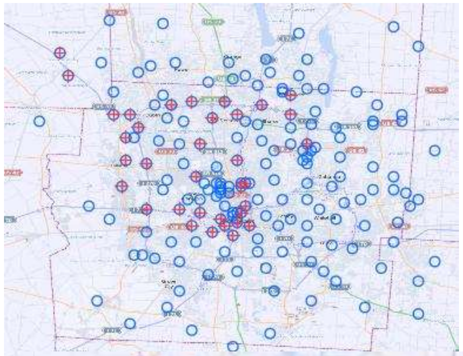




NN-K-SVD



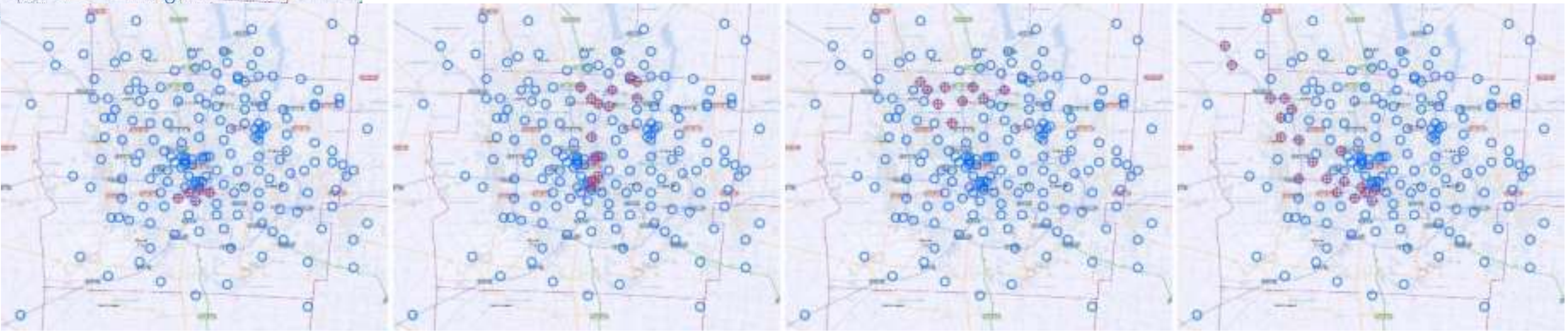
Comparison between local and collaborative DL



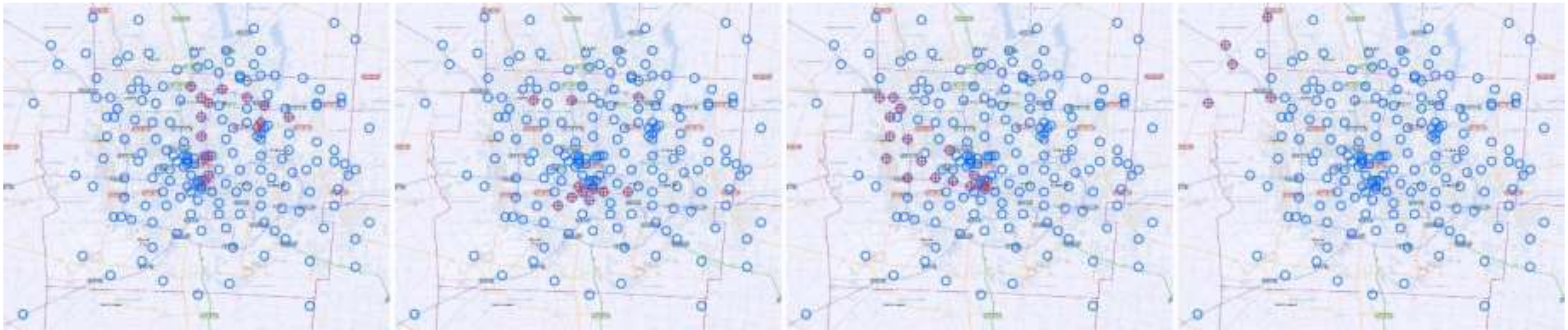
test path

May 11

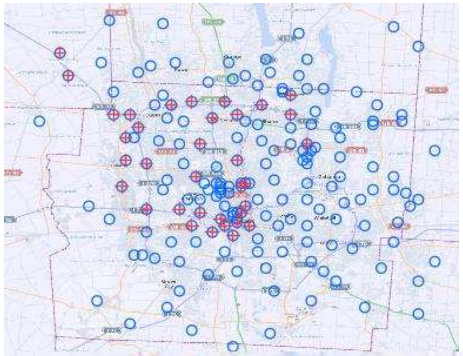
associated atoms for local NN-K-SVD at node 2



associated atoms for local NN-K-SVD at node 5



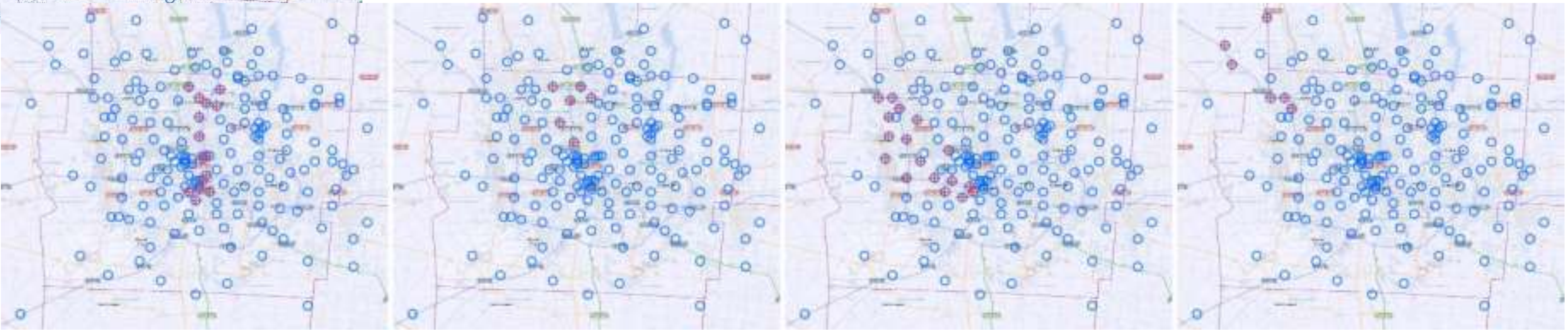
Comparison between local and collaborative DL



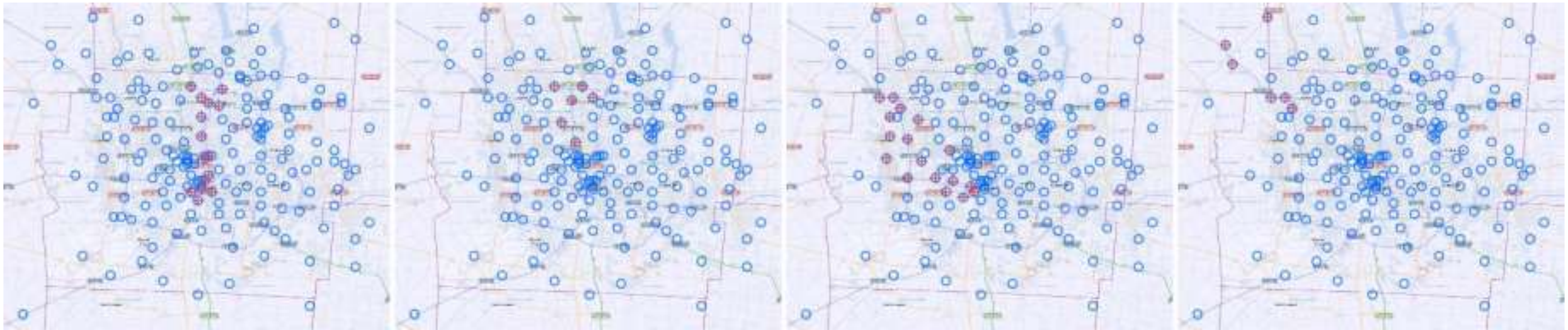
test path

May 11

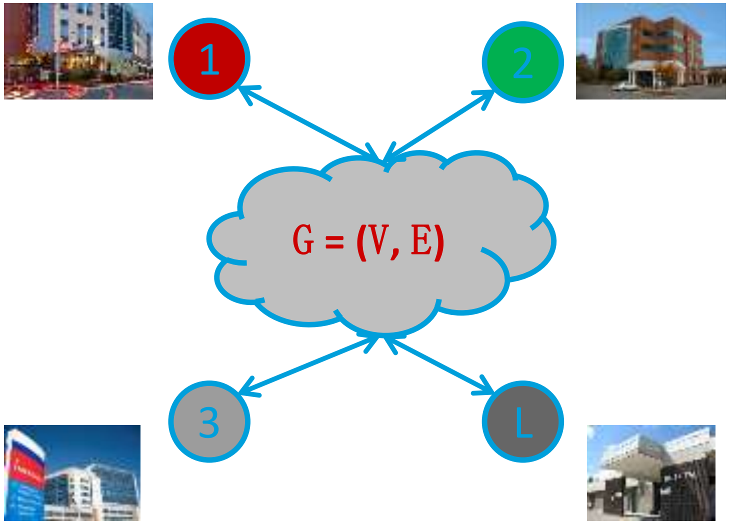
associated atoms for cloud NN-K-SVD at node 2

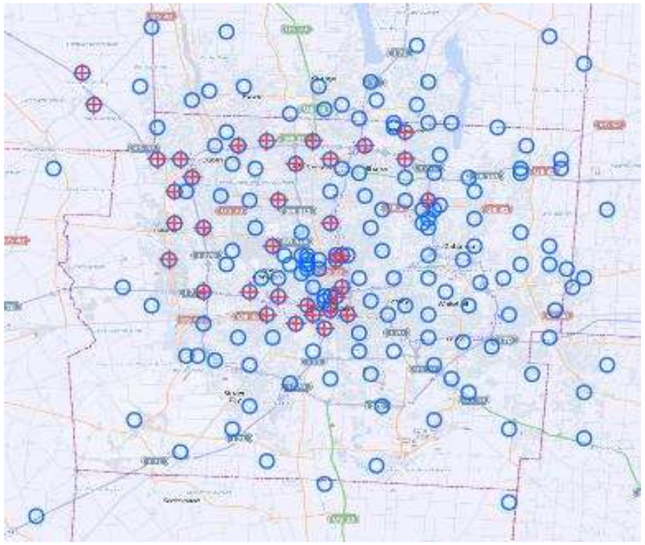


associated atoms for cloud NN-K-SVD at node 5



Conclusions





- Introduced dictionary learning problems

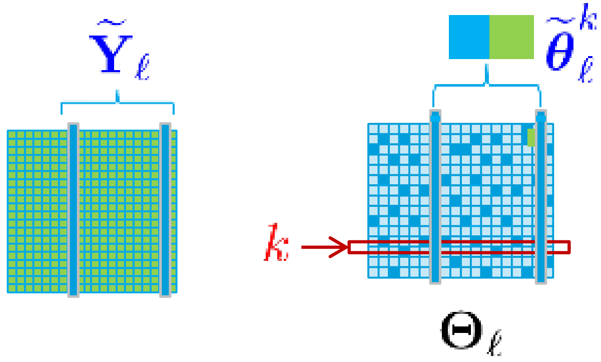
- Discussed in detail the cloud NN-K-SVD algorithm

- Empirical validation of centralized/distributed dictionary learning algorithms using slippy tile data for learning mobility patterns

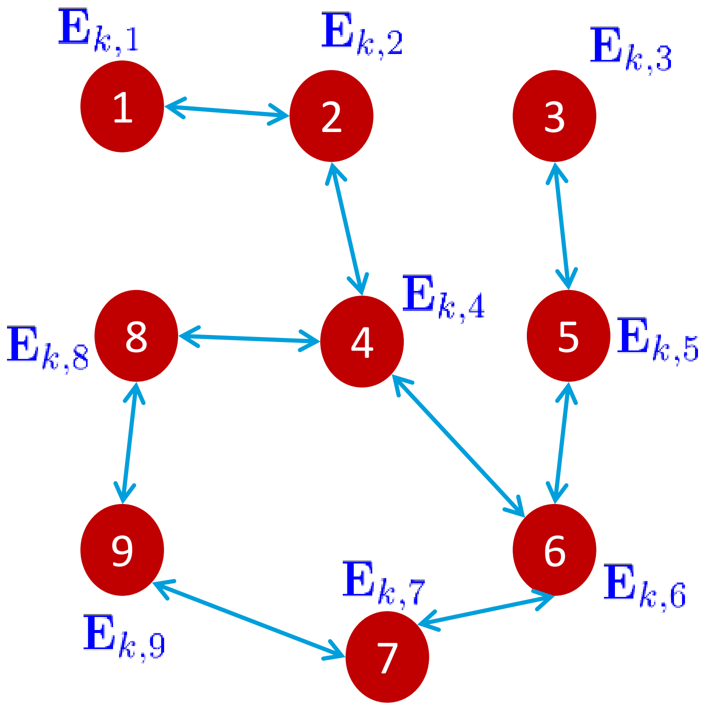
Thank You

RUTGERS AT&T

Collaborative dictionary update in cloud NN-K-SVD


$\widetilde{\mathbf{Y}}_\ell$
$\widetilde{\boldsymbol{\theta}}_\ell^k$
k
$\boldsymbol{\Theta}_\ell$

$$\mathbf{E}_{k,\ell} = \widetilde{\mathbf{Y}}_\ell - \sum_{j \neq k} \mathbf{d}_{\ell,j} \widetilde{\boldsymbol{\theta}}_\ell^j$$



Challenge 1: Estimate principal singular vector of $\boldsymbol{E}_k$ collaboratively when we only know $\boldsymbol{E}_{k,\ell}$ at each site?

Solution ⇨ Distributed version of power method

Challenge 2: How to distribute the following two steps?

$$\boldsymbol{d} = \frac{\boldsymbol{E}_k \boldsymbol{g}^T}{\boldsymbol{g}\boldsymbol{g}^T}, \qquad \boldsymbol{g} = \frac{\boldsymbol{d}^T \boldsymbol{E}_k}{\boldsymbol{d}^T \boldsymbol{d}}$$

Solution ⇨ Consensus averaging

$$\mathbf{E}_k = \begin{bmatrix} \mathbf{E}_{k,1} & \mathbf{E}_{k,2} & \cdots & \mathbf{E}_{k,L} \end{bmatrix}$$