MCMC strategies for Task-Specific Region Partition Problem

Hua Wei
College of IST, Penn State University

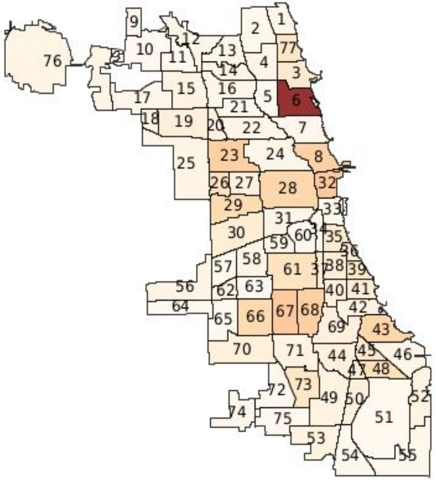
Learning Task-Specific City Region Partition

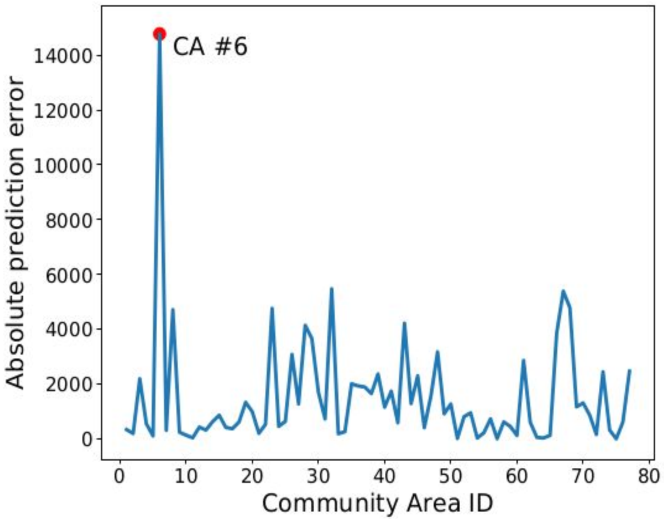
Task: Crime Prediction

Given:
- tract features and crime number in 2010
- tract features in 2011
- Communities consist of tracts

Task:
predict crime number for every community in 2011

Easy solution:
- Aggregate tract-level data into community-level features and crime number as training data
- Fit regression model





Left: crime prediction error at community level. Right: community #6 is an outlier.

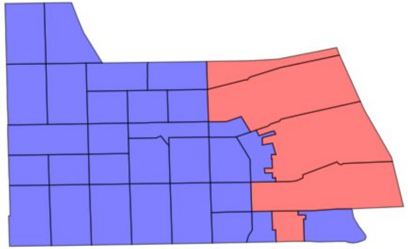
Learning Task-Specific City Region Partition

Explain the crime prediction error.

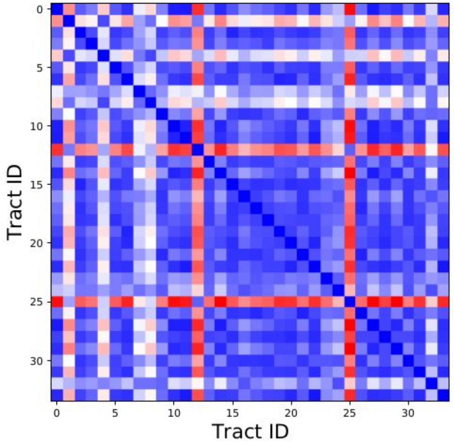
The eastside of Community #6 is different from others.

Question:

How do we obtain appropriate partitions?



(a) Census tracts of community #6.



(b) Tract similarity.

Appropriate partitions are needed!

- **Misalignment problem**: different observations are not in the same scale.
    - *Crime reports at point level, demographics at block level.*
- The existing administrative boundaries are **human-defined** and **static.**
    - *Chicago administrative boundaries are defined by university scholars and keep being used for over 100 years.*
- **Task dependent partition**: for different prediction tasks, the partition may not be the same.
    - *In State College the Beaver stadium is a hotspot for traffic but not for crime incidents.*

Problem definition: task-specific region partition

Input:

- Target property observed at tract level: ***y***.
- Other properties (e.g., demographics) available for all tracts: ***x***.
- Spatial adjacency of all tracts: **G**.

Output:

- Partition tracts into community areas with **Y** and **X**, such that

$$\underset{\mathcal{Z},f}{\arg\min} \sum_{j=1}^{m} \left(\boxed{||Y_j - f(X_j)||_2} + \boxed{G(Z_j)} \right) \quad \text{(Eq. 1)}$$

Task prediction error

Constraint on partition, e.g. variance of population

$\mathcal{Z}$: partition. (mapping from tracts to community)

f: task function on community (regression model, etc.)

Given a fixed partition $\mathcal{Z}$, the optimal task function f can be easily learned.

The challenge lies in searching through the partition space.

A partition over City $\boldsymbol{T}$ is denoted as $\boldsymbol{Z} = \{Z1, Z2, \cdots, Zm\}$, satisfying the following:
(1) (subset) $\forall j, Zj \subset T$; (2) (non-overlapping) $\forall p,q, Zp \cap Zq = \varnothing$; (3) (completeness) $\boldsymbol{U\{Z j\} = T}$;
(4) (spatial-continuity) $\forall j, Pj$ defines a polygon with exact one connected component.

This problem is challenging

1. It's difficult to calculate the maximum likelihood of objective function.
    a. Gradients on discrete variables are non available.
    b. Region properties (features and target variable) and model coefficients are high dimensional
    c. region properties change simultaneously when we change the region partition
    d. Complicated constraints should be met.
2. It is a NP-hard *combinatorial optimization* problem.
    a. Brute Force Solution: Enumerate all the possible combinations
        - O(M^N) - M is the number of tracts, N is the number of communities

Solution: Techniques like Simulated Annealing to approximate global optimization

- Approximate the global optimum with the Markov Chain Monte Carlo (MCMC) method

Markov Chain Monte Carlo (Metropolis-Hastings)

Motivation: Obtaining a sequence of random samples from a distribution for which direct sampling is difficult. It constructs a Markov chain that will ultimately converge through stochastic sampling. We only **care about the last state**.

In our case:

- state space is all possible partitions *Z*
- quality measure $\mathcal{F}(Z)$, a partition Z with lower value is more likely to be optimal
- proposal function ***q(Z'|Z)***, defines the transition probability
- ***p(Z)*** is the Boltzmann distribution $p(Z) = \frac{e^{-\mathcal{F}(\mathcal{Z})/T}}{P}$
- Stopping criteria:
  - Number of iterations/the standard deviation of $\mathcal{F}(Z)$

Algorithm 1 MCMC method to search $\mathcal{Z}$.

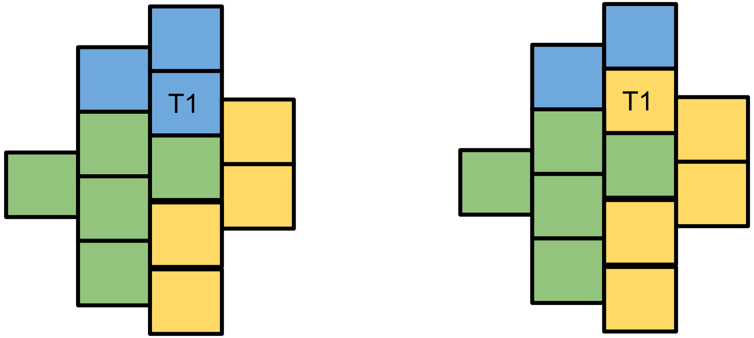
$\mathcal{Z} \leftarrow \mathcal{Z}_0$
while not satisfies stopping criteria **do**
  Sample $u \leftarrow \mathcal{U}_{[0,1]}$
  Sample $\mathcal{Z}' \leftarrow q(\mathcal{Z}'|\mathcal{Z})$
  $\gamma = \min\left[1, \frac{p(\mathcal{Z}')q(\mathcal{Z}|\mathcal{Z}')}{p(\mathcal{Z})q(\mathcal{Z}'|\mathcal{Z})}\right]$
  if $u < \gamma$ **then**
    $\mathcal{Z} \leftarrow \mathcal{Z}'$
  end if
end while

Naive MCMC

- Start from administrative boundary
- **Randomly** select one tract to **flip its assignment**.
- Decide whether to accept the new partition with training error on prediction task f.

$$q(\mathcal{Z}'|\mathcal{Z}) = q(\mathcal{Z}|\mathcal{Z}') = \frac{1}{|\mathcal{T}_b| \cdot |Adjacent(t_i)|}$$

$$\gamma = \min[1, \frac{p(\mathcal{Z}')}{p(\mathcal{Z})}]$$



Flip the assignment of a tract.

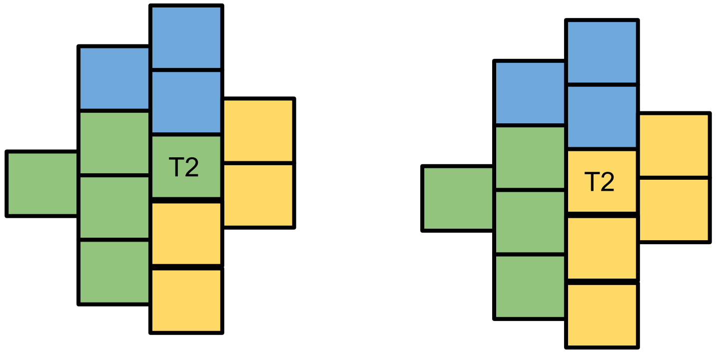
This could last a long time - since it's randomly select one tract to flip

Guided MCMC - Softmax MCMC

- Start from administrative boundary
- **Select tract from Community with the highest prediction error** (according to a softmax function), then **flip its assignment**.
- Decide whether to accept the new partition with training error on prediction task f.

$$Q(Z_j) = \frac{\exp(||Y_j - f(X_j)||_2)}{\sum_{k=1}^{m} \exp(||Y_k - f(X_k)||_2)}$$

$$q(\mathcal{Z}' | \mathcal{Z}) = \frac{Q(Z_j)}{|Z_j \cap T_b| \cdot |Adjacent(t_i)|}$$

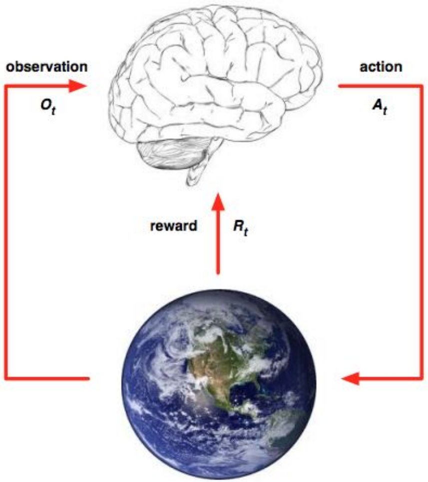


Flip the assignment of a tract *T2*.

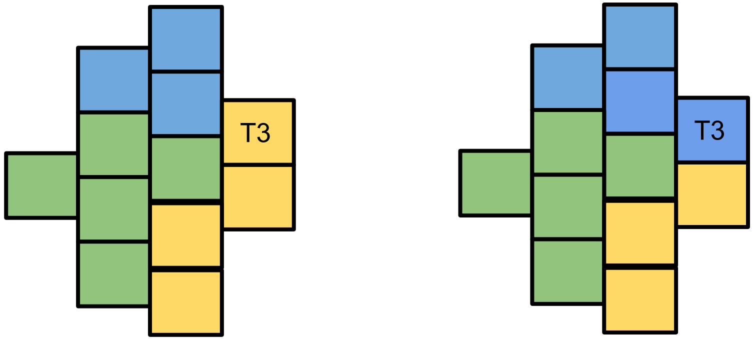
Tract *T2* is selected because the green community has the highest prediction error.

MCMC with reinforcement learning (RL)

- Start from administrative boundary
- **Instead of pre-define sampling strategy, learn where to sample with RL**, then **flip its assignment**.



At each step t the agent:
- Executes action A_t
- Receives observation O_t
- Receives scalar reward R_t



Tract *T3* is selected because based on past experience, T3 is more likely to decrease $\mathcal{F}$ most

MCMC with reinforcement learning (RL)

- Start from administrative boundary
- **Instead of pre-define sampling strategy, learn where to sample with RL**, then **flip its assignment**.

Challenges:

- Huge state space
- Large and dynamic action space
- Training overhead is high

Learning

$$Q(\mathcal{Z}, \langle t^k, Z^k \rangle) = \Delta \mathcal{F} + \delta \cdot \sum_{a \in \{\langle t^{k+1}, Z^{k+1} \rangle\}} Q(\mathcal{Z}', a)$$

Q is a neural network
δ is set to 0

$\mathcal{Z}$: current partition state
<t, Z> : action (a tract to flip)

- Sampling: $\arg\max_{\langle t, Z \rangle} Q(\mathcal{Z}, \langle t, Z \rangle)$

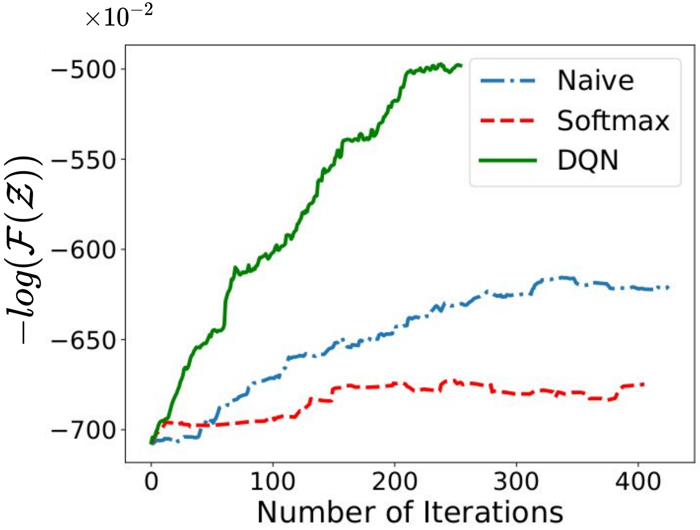
Take the max of 30 random samples

Results

Method	MAE	
	Crime	House price
Admin	1715.91	31.29
Agglomerative	72201.00	50.34
K-means	2887.83	32.40
Spectral	1440.57	29.66
Naive	1073.42(81.93)	25.73(2.76)
Softmax	1041.68(76.75)	27.13(2.98)
DQN	**746.13**(154.19)	**25.16**(1.30)

MAE of two prediction tasks.

Clustering first, then do prediction

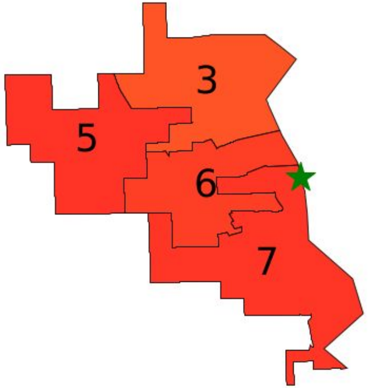


Convergence of three proposed methods.

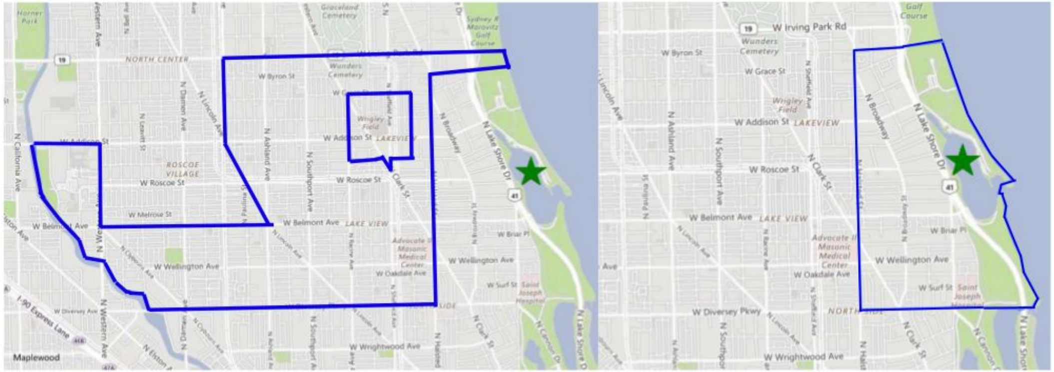
Dataset:
[1] https://data.cityofchicago.org/
[2] https://www.zillow.com/

Case study: Community #6

Community #6 from administrative boundary



DQN partition around Community #6



Zillow's self-defined region (Zillow.com: real estate website)

The green star marks the location of **Belmont Harbor** (one of Chicago's largest boating areas).

DQN community #7: contain leisure destinations such as the Lincoln Park Zoo, and numerous beach areas.

Suggestions?