Comparison of Traditional and Constrained Recursive Clustering Approaches for Generating Optimal Census Block Group Clusters

Damon Gwinn[1], Jordan Helmick[2], Natasha Kholgade Banerjee[1], and Sean Banerjee[1]

[1] Clarkson University, Potsdam, NY
{gwinndr,nbanerje,sbanerje} @clarkson.edu
[2] MedExpress, Morgantown, WV
jordan.helmick@medexpress.com

Abstract. Census block groups are used in location selection to determine the average drive time for all residents within a given radius to a proposed new store. The United States census uses 220,334 block groups, however the spatial distance between neighboring block groups in densely populated areas is small enough to cluster multiple block groups into a single unit. In this paper, we evaluate the efficiency and accuracy of drive time computations performed on clusters generated by our novel approach of constrained recursive reclustering as run on three traditional clustering algorithms—affinity propagation, *k*-means, and mean shift. We perform comparisons of our constrained recursive reclustering approach against drive times computed using the original census block group, and using clusters obtained by traditional reclustering. Unlike traditional clustering, where clustering is performed in a single pass, our approach continues reclustering each new cluster until a user specified stopping criteria is reached. We show that traditional clustering techniques generate sub-optimal clusters, with large spatial distances between the cluster centroid and cluster points making them unusable for computing drive times. Our approach provides reductions of 81.2%, 83.4%, and 10.2% for affinity propagation, *k*-means, and mean shift respectively when run on 220,334 census block groups. Using 200 randomly sampled locations each from Lowe's, CVS, and Walmart, we show that compared to the original block groups there is no statistically significant difference in drive time computations when using clusters generated by constrained recursive reclustering with affinity propagation for any of the three businesses, and with *k*-means for CVS and Walmart. While statistically significant differences are obtained with *k*-means for Lowe's and with mean shift for all three businesses, the differences are negligible, with the mean difference for each location set being within 30 seconds.

Keywords: Location selection, census block group, affinity propagation, k-means, mean shift, constrained, recursive, clustering

1 Introduction

Location selection is used to analyze the feasibility of a new location, such as a new medical facility or retail store, by comparing the number of potential customers and their average drive time to those of competing locations. A new location is less likely to

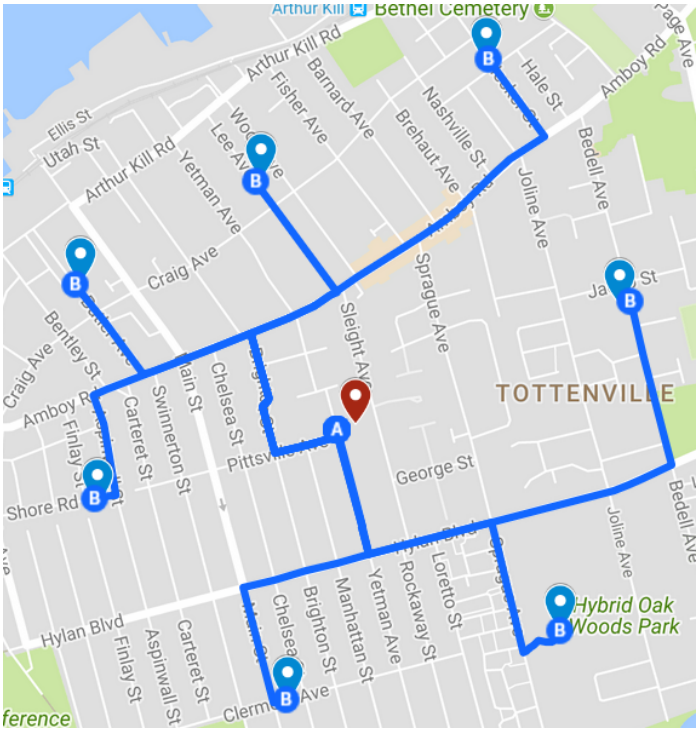


Fig. 1. Tottenville, Staten Island, NY is described by 8 block groups. The distance from the block group labeled A to all block groups labeled B is 1 mile, with a drive time between 2-3 minutes.

succeed if it is too distant from the customer base or is out-positioned by a competitor. Census block groups are used to compute drive times for potential customers, as computing drive times for each individual resident is infeasible. The United States 2010 Census utilizes 220,334 block groups, with each block group representing between 600 and 3,000 people [9].

To determine the feasibility of a given location, companies need to evaluate thousands of potential locations and their associated competitors. With major retailers opening hundreds of new locations annually, computing drive times for each potential location from census block groups can become computationally infeasible. For example, Walmart has opened on average 443 new locations worldwide every year for the past 10 years [33]. In Gwinn et al. [18], we presented a constrained recursive cluster splitting technique for generating census block group clusters using affinity propagation as the clustering algorithm. We provided a 5$\times$ speed up in the drive time computation process by reducing the number of block groups from 220,334 to 41,442 clustered block groups. Our approach showed no statistically significant difference in drive times when evaluated on a sample of 200 random geographic locations placed within the continental United States and 300 random Walmart locations across the United States. Our recursive reclustering technique is based on the fact that neighboring block groups in densely

populated areas show small spatial distances. As shown in Figure 1, densely populated areas, such as Tottenville, Staten Island, NY, are comprised of multiple block groups. In the figure, block groups labeled B are at a distance of 1 mile, with a drive time between 2 to 3 minutes, from the block group labeled A. Our approach provides computational speed ups by considering block group A to be representative of all block groups A and B, thus reducing the number of drive time computations for a new location from 8 to 1.

In this paper, we compare the performance of three traditional clustering techniques—affinity propagation, k-means and mean shift—to our novel constrained recursive reclustering approach which can use any of the three clustering algorithms as to generate initial clusters in the base case. We show that traditional clustering techniques generate sub-optimal clusters that are unusable for computing drive times for location selection due to the large mean distances from the cluster centroid to the cluster points. We show the robustness of our recursive reclustering technique by demonstrating that both recursive affinity propagation and recursive k-means provide a similar reduction in drive time. We validate our approach by computing exact drive times for 200 random Lowe's Home Improvement, 200 random CVS Pharmacy, and 200 random Walmart locations within the United States. We show that recursive affinity propagation shows no statistically significant difference in drives times for all three datasets. While recursive k-means shows a statistically significant difference in drive times for Lowe's, the difference in drive times is under 30 seconds and is insignificant to the consumer. Recursive mean shifts shows a statistically significant difference in drive times for all three datasets, however the differences are practically insignificant to the consumer and under 30 seconds.

The rest of this paper is organized as follows: we discuss the work related to our approach on site analysis in Section 2. We describe our approach on constrained recursive reclustering in Section 3, together with the recursive splitting of clusters, the clustering algorithms analyzed, and the constraint set used for splitting. Section 4 provides our method to evaluate the efficiency of our approach by computing drive times within a bounding box centered around a site location. In Section 5, we describe our dataset, and we show the computational improvements gained by performing constrained recursive reclustering of block groups. We discuss the practical and statistical differences in drive times using 200 random locations each from CVS Pharmacy, Lowe's Home Improvement, and Walmart in Section 6 for the original block groups, clusters obtained by traditional clustering, and clusters obtained by constrained recursive reclustering. We provide a summary and directions for future research in Section 7.

2 Related Work

Our work falls in the category of techniques that use census data to perform site analysis and selection. A number of approaches use census data to study the creation or propagation of food deserts. Blanchard and Lyson [4] use zip code business pattern data from the U.S. Census Bureau to determine the effect of supercenter grocery stores in reducing access to low-cost grocery stores for disadvantaged rural populations. The approach of Farber et al. [12] performs analysis of travel time from each census block in Cincinnati, Ohio to nearest supermarkets at different times of the day to investigate

the occurrence of food deserts. Similarly, the approach of Jiao et al. [19] identifies food deserts by using census data to combine income with computation of travel times to supermarkets using walking, bicycling, ride transit, or driving within 10 minutes.

In the area of emergency care access, Branas et al. [6] use census block group data in conjunction with trauma centers and base helipads, and estimate that 69.2% and 84.1% of all U.S. residents have access to level I or II trauma centers within 45 and 60 minutes respectively. They estimate drive time using mathematical models that use p-norms to approximate the driving distance [25] together with drive speed estimates in urban, suburban, and rural categories. They obtain drive speed estimates by categorizing average population densities in each block group as high for urban, medium for suburban, and low for rural, and they add extra time to account for receipt of emergency call and time spent at the scene of emergency. Carr et al. [7] use emergency department (ED) location information from the National Emergency Department Inventories and the drive time estimation approach of Branas et al. [6] to determine that 71% of the U.S. population has access to an ED within 30 minutes, and 98% within 60 minutes. The approach of Nattinger et al. [27] uses the haversine formula to calculate the driving distance between hospitals offering radiotherapy services and census tracts for U.S. patients with breast cancer between 1991 and 1992. Simple distance estimation approaches using p-norm functions or haversine formula to approximate the distance between two locations do not take into account the effect of road structure and local changes in speed.

The approach of Athas et al. [2] uses the ArcGIS software to estimate shortest drive times between radiation treatment facilities and female individuals diagnosed with breast cancer in New Mexico between 1994 and 1995. While 70% of individuals were geocoded to a unique street address, the remaining 30% who had post office box or rural addresses were geocoded to the centroid of the ZIP code. Similarly, the approach of Goodman et al. [16] uses digitized road maps together with road category and traffic weightings to compute distances between the geographic centers of zip codes and nearest hospital and primary care physicians. Zip code centers are often not ideal for drive time calculations, as they may not provide optimal drive time estimates for individuals living at zip code boundaries. Instead of relying on zip code centers, our approach uses constraints on distance estimates and population counts to estimate centroids representative of block groups in urban or rural areas for more optimal drive time calculation.

The approach of Nallamothu et al. [26] obtains drive time estimates by using data on interstate, state, and local roads in Topologically Integrated Geographic Encoding and Referencing (TIGER) data from 2000 to estimate drive distances together with Census Feature Classification Codes for each road type to determine drive speeds. They use drive time estimates and population data to determine that 79% of adult population of age 18 and older lives within 60 minutes of a hospital that performs percutaneous coronary intervention. While their approach uses higher resolution data compared to Branas et al. [6] and Carr et al. [7] in estimating drive time, the high-resolution data introduces a higher drive time computation time, which can become intractable for repeated testing of potential new sites. Our approach resolves the issue of intractability by clustering census block groups into smaller clusters. Li et al. [23] use a map-matching algorithm to map taxicab GPS trajectories onto a road network. They use partitioning around k-medoids to iteratively solve the problem of finding the nearest locations to respond to

emergency requests. While they use the k-medoids clustering algorithm [28], their clustering is not meant to reduce the quantity of data used in drive time computation as in our approach. Instead, they use the update step of k-medoids clustering to iteratively converge to the best response locations for emergency requests.

While our work focuses on using census data and drive time for site analysis, several approaches on site selection integrate a variety of external features in determining the quality of a site. Xu et al. [40] use features such as distances to the city center, traffic, POI density, category popularity, competition, area popularity, and local real estate pricing to determine the feasibility of a location. The approach of Karamshuk et al. [21] uses features mined from FourSquare along with supervised learning approaches to determine the optimal location of a retail store. A number of approaches use information obtained from social media platforms, such as popularity based on user reviews [37] or based on number of user check-ins and viability of location as obtained from Twitter and FourSquare [21, 30, 44, 43, 38, 10]. Given the large number of features that may be used to evaluate a site, some approaches use fuzzy techniques to determine sites that show the best compromise between various site selection criteria [20]. Approaches have used fuzzy techniques to determine the optimal number of fire stations at an airport [34], and optimal locations of new convenience stores [22] and factories [8, 42]. There also exist approaches that use user expertise to weight location selection criteria in reaching a compromise on best location [35, 41, 1].

While our approach is designed to provide drive time computations for user-provided site queries, there exist approaches that perform automated determination of an optimal site query given a set of existing sites and current customer locations [39, 14]. The approach of Ghaemi et al. [15] performs optimal query estimation while addressing issues caused by movement of existing sites and customers. Banaei et al. [3] perform reverse skyline queries to incorporate additional criteria such as distance to location and distance to competitors in optimal location query estimation.

3 Constrained Recursive Reclustering

Our approach on recursive reclustering uses a user-provided constraint set to recursively split an initial set of clusters into a final set that satisfies the constraint set. Our approach is flexible enough to accommodate any form of clustering algorithm, and can use constraint sets defined using operations such as intersections, unions, and complements on a number of inequality constraints. We discuss the recursive cluster splitting approach in Subsection 3.1. The clustering algorithms analyzed in this work are discussed in Subsection 3.2, while the particular constraint set used in this work is discussed in Subsection 3.3. Unlike existing approaches on constrained clustering that address the issue of satisfying user-defined constraints and user-provided parameters such as number of clusters [5, 36], our approach focuses on addressing user-defined constraints for non-parametric clustering, i.e., in our approach, the user does not need to provide parameters such as the number of k-means clusters or the mean shift kernel bandwidth.

3.1 Recursive Cluster Splitting

For any particular clustering algorithm discussed in Subsection 3.2, our reclustering approach recursively performs cluster splitting approach starting from an initial set of clusters generated using the approach discussed in Subsection 3.2 for each algorithm. In addition to the initial clusters, our approach uses a constraint set Ω that can be defined for each cluster using operations such as intersections, unions, and/or complements on a number of inequality constraints. Starting from the initial cluster set, our approach splits each cluster into a set of child clusters using the particular clustering algorithm if the constraint set Ω is not met. The clustering approach is performed recursively till Ω is met for each resultant cluster.

Algorithm 1 summarizes the steps of our recursive clustering approach. The algorithm adapts the recursive clustering splitting algorithm from our prior work [18] to work with any clustering algorithms and constraint sets. The initial clustering algorithm runs in $O(tn^2)$ time and produces k clusters, where t represents the number of iterations until convergence and n represents the number of samples. Each of the k clusters is reclustered in $O(tm_i{}^2)$ time, where m_i represents the number of points in the i^{th} cluster and $i \in [1, 2, \cdots, k]$.

While our recursive reclustering approach is related to divisive hierarchical clustering techniques [31], it differs from them in that hierarchical clustering approaches focus on generating clusters that meet distance constraints intrinsic to the cluster points, while our approach integrates external user-defined constraints. As discussed in Section 5, we compare the effect of our recursive reclustering approach on drive time computation to using the original block groups, and to using traditional clustering as obtained by running the clustering algorithms in Subsection 3.2 without reclustering.

3.2 Clustering Algorithms Analyzed in this Work

In this paper, we compare the performance of three different clustering algorithms—affinity propagation [13], k-means [24], and mean shift [11].

Affinity Propagation. The approach of affinity propagation, proposed by Frey and Dueck [13], uses message passing between data points to select a set of exemplar points. Each exemplar point is representative of a cluster of data points in its vicinity. The advantage of the method is that it does not require user-specified parameters such as the number of clusters as in k-means or the kernel bandwidth as in mean shift. The method iteratively refines estimates on exemplar points by updating responsibility messages that data points send to candidate exemplar points on the suitability of the exemplar points to represent the data points, and availability messages that candidate exemplar points send back to data points to reflect how well the candidate exemplar represents each data point based on accumulated responsibility evidence from other data points. In this work, we use the affinity propagation function implemented in the `scikit-learn` toolbox [29], with a damping factor of 0.9 to reduce numerical oscillations in updates of the responsibility and availability. We use 2000 maximum iterations, and 200 convergence iterations, i.e., iterations over which the number of clusters remain consistent for convergence.

The affinity propagation algorithm does not directly accommodate external user-defined constraints. We find that in our work, the clusters generated by affinity propagation are large and do not satisfy the constraint set discussed in Subsection 3.3. We use the recursive clustering approach discussed in Subsection 3.2 to perform further clustering using affinity propagation till the constraint set is met.

k-means. We use the k-means function in the `scikit-learn` toolbox that implements Lloyd's algorithm [24] to partition the region in Euclidean space containing the locations of the points into a user-specified number of clusters, k. Given an initialization for the cluster centroids, Lloyd's algorithm iteratively computes the Voronoi diagram for the k clusters, and updates the centroid given each Voronoi cell. The computation of the Voronoi diagram effectively assigns each sample to its nearest centroid. The algorithm proceeds till the k-means objective function defined as the within cluster sum-squared Euclidean distance converges, i.e., till the difference between consecutive values of the function falls below a tolerance level. Since changes in the initial centroid locations yield different values of the objective function, we use m initializations of the k-means algorithm, and select the initialization that yields the lowest value of the k-means objective function. In this work, we use the default value of $m = 10$ in the `scikit-learn` toolbox, together with the default values of 0.001 for tolerance level and 300 for maximum number of iterations.

The principal challenge of the k-means approach is that the number of clusters k needs to be pre-specified *a priori*. In this work, we use the approach of silhouettes [32] to select the best number of clusters between 2 and an upper bound K on the number of clusters. For a candidate number of clusters k, the silhouette score is obtained as the difference between the mean nearest-cluster Euclidean distance b and the mean intra-cluster Euclidean distance a, scaled by the maximum value of a and b to obtain a value between -1 and 1. Values near 1 indicate good clusters, values near 0 indicate overlapping clusters, while values near -1 indicate that a point is incorrectly assigned to a cluster. We conduct k-means for $k \in [2, K]$, and select the best value of number of clusters $k^\star$ as the k for which the silhouette score is highest.

While one choice for the value of K is the total number of data points n, running k-means and silhouette score calculation for $k \in [2, n]$ is computationally intensive. We perform silhouettes-based k-means clustering using $k \in [2, n]$ for a random sampling of 25% U.S. states. We find that all U.S. states in the random sampling have a value of $k^\star \leq 60$, while 70% of the U.S. states in the random sampling have a value of $k^\star \leq 20$. The constraint set Ω discussed in Subsection 3.3 is not met for all clusters in any of the U.S. states, indicating that recursive clustering is necessary with silhouette-based k-means. To optimize between run-time and optimal selection of number of clusters, we set K to 20 for all 50 U.S. states in the initial and recursive clustering steps.

Mean Shift. The mean shift approach for clustering [11] is a mode-seeking algorithm that recursively updates the means of a collection of shifted kernel functions in representing the maxima or modes or a density function. Each cluster provided as the result of the mean shift clustering algorithm is given as the set of data points closest to the mean of the kernel function representing that cluster. In this work, we use radial basis function (RBF) kernels to perform mean shift as implemented in the `scikit-learn`

toolbox. While the mean shift approach is considered a non-parametric technique in that it does not require specification of the number of clusters, it does require specifying the bandwidth of the kernel used in mode-seeking. Higher bandwidth kernels yield larger clusters, while lower bandwidth kernels yield smaller clusters. We use the bandwidth estimation function built into `scikit-learn` that uses the 30^{th} percentile of pairwise distances as the band-width.

3.3 Constraint Set Used in this Work

The constraint set Ω in this work is defined as the union between a cluster points count constraint ω_c and the intersection of a population count constraint ω_p and a distance constraint ω_p, i.e.,

$$\Omega = \omega_c \cup (\omega_d \cap \omega_p). \tag{1}$$

This induces our approach to perform recursive clustering till either the number of points in the cluster falls below a user-provided threshold, i.e., till the cluster points count constraint ω_c is satisfied, or till the total population and the mean haversine distance between the cluster points to the cluster centroid both fall below user-provided thresholds, i.e., till both ω_p and ω_d are satisfied.

The cluster points count constraint, ω_c for a particular cluster c is defined as

$$\omega_c : |I_c| \leq c_{\text{bound}}, \tag{2}$$

where I_c represents the index set to the number of points in the cluster c, $|I_c|$ represents the size of I_c or the count of the number of points, and c_{bound} represents the user-provided upper bound on the cluster count.

The population count constraint, ω_p for cluster c is defined as

$$\omega_p : p_c \leq p_{\text{bound}}, \tag{3}$$

where p_c represents the user population in that cluster as obtained by summing the population counts of all block groups in cluster c, while p_{bound} represents the user-provided upper bound on the population count.

The distance constraint, ω_d for cluster c is defined as

$$\omega_d : \frac{1}{|I_c|} \sum_{i \in I_c} \left(2R \operatorname{atan2}\left(\sqrt{a_i}, \sqrt{1-a_i} \right) \right) \leq d_{\text{bound}}, \tag{4}$$

where the term on the left hand side of the inequality represents the mean haversine distance between the cluster centroid and the cluster points indexed by the set I_c, and d_{bound} represents the user-provided upper bound on the mean haversine distance. In Inequality (4), a_i represents the haversine of the central angle between each point represented by its latitude ϕ_i and longitude λ_i to its cluster centroid represented by ϕ_c and λ_c, and is computed as

$$a_i = \sin^2 \frac{\phi_c - \phi_i}{2} + \cos\phi_i \cdot \cos\phi_c \cdot \sin^2 \frac{\lambda_c - \lambda_i}{2}. \tag{5}$$

For all results in this work, the distance d_{bound} is set to 5 miles, the population count bound p_{bound} is set to $20,000$, and the cluster points count bound c_{bound} is set to 10 points.

Algorithm 1: Recursive Cluster Splitting

Input: Sets of latitudes and longitudes for initial cluster points
$\{\{(\phi_i, \lambda_i) : i \in I_{c_{\text{init}}}\} : c_{\text{init}} \in \mathcal{C}_{\text{init}}\}$,
Set of latitudes and longitudes for initial cluster centroids $\{(\phi_{c_{\text{init}}}, \lambda_{c_{\text{init}}} : c_{\text{init}} \in \mathcal{C}_{\text{init}}\}$,
and user-provided bounds c_{bound}, d_{bound}, and p_{bound}
Output: Set of final clusters, O

for $c_{init} \in \mathcal{C}_{init}$ **do**
  $P_{c_{\text{init}}} \leftarrow \{(\phi_i, \lambda_i) : i \in I_{c_{\text{init}}}\}$
  O = `split` $(\mathcal{P}_{c_{init}}, O)$
  return O
end
Procedure `split` $(\mathcal{P}_c, O)$
  Compute the constraint set Ω using Equation (1)
  if *the constraint set* Ω *is not met* **then**
    Split cluster represented by points in $\mathcal{P}_c$ by clustering them into smaller clusters $\{P_{\bar{c}} : \bar{c} \in \mathcal{C}_c\}$ using the clustering algorithm
    for $\bar{c} \in \mathcal{C}_c$ **do**
      return `split` $(\mathcal{P}_{\bar{c}}, O)$
    end
  else
    $O \leftarrow O \cup \mathcal{P}_c$
    return O
  end

4 Drive Time Computation

We evaluate our approach by performing drive time computation from site locations for a particular business to all points within a bounding box of a user-specified half-size centered at the business location. The points refer to either the original block groups or the cluster centroids obtained using recursive reclustering on the clustering algorithms discussed in Subsection 3.2, and represent customers most likely to visit the business location. Given a bounding box half-size d, and the location of the business represented by latitude ϕ_1 and longitude λ_1, we use the inverse haversine formula to obtain the north-east and south-west locations of the bounding box, denoted by latitudes ϕ_{max} and ϕ_{min} and longitudes λ_{max} and λ_{min} respectively, as described in Algorithm 2 and as discussed in our prior work [18]. We set the bounding box half-size d to 5 miles. As a note, while we use the haversine distance formula as a heuristic to compute the bounding box and to split the clusters as part of the constraint set in Section 3.3, we do not use the haversine distance between the starting and ending points to compute the drive times as in the approach of Nattinger et al. [27], since the start-to-end haversine distance does not account for the road structure and changing speed limit between the two locations. Instead, we use the Google Maps Distance Matrix API [17] to generate drive times from the business location to all points within the bounding box. Algorithm 1 generates the bounding box within $O(1)$ time, while the drive time computations using Google Maps API are performed in $O(l)$ time, where l is the number of points in the bounding box.

Algorithm 2: Bounding Box Computation as performed in Gwinn et al. [18]

Parameters: *MINLAT* (min latitude):-90°,
MAXLAT (max latitude):90°,
MINLON (min longitude):-180°,
MAXLON (max longitude):180°,
R (radius of earth): $6,371$ km.

Input: Distance d and location (ϕ_1,λ_1)

$\phi = \frac{d}{R}$
$\phi_{\min} = \phi_1 - \phi$
$\phi_{\max} = \phi_1 + \phi$
if $\phi_{min} > MINLAT \wedge \phi_{max} < MAXLAT$ **then**
  $\lambda = \sin^{-1}\left(\frac{\sin\phi}{\cos\phi_1}\right)$
  $\lambda_{\min} \leftarrow \lambda_1 - \lambda$
  if $\lambda_{min} < MINLON$ **then**
    $\lambda_{\min} \leftarrow \lambda_{\min} + 2\pi$
  end
  $\lambda_{\max} \leftarrow \lambda_1 + \lambda$
  if $\lambda_{max} > MAXLON$ **then**
    $\lambda_{\max} \leftarrow \lambda_{\max} - 2\pi$
  end
else
  $\phi_{\min} \leftarrow \max(\phi_{\min}, \text{MINLAT})$
  $\phi_{\max} \leftarrow \min(\phi_{\max}, \text{MAXLAT})$
  $\lambda_{\min} \leftarrow \text{MINLON}$
  $\lambda_{\max} \leftarrow \text{MAXLON}$
end

Our approach of using clustering accelerates the drive time calculations in comparison to the original block groups due to the $O(l)$ dependence of the drive time queries.

5 Traditional vs. Recursive Reclustering Results

We compare traditional clustering to recursive reclustering to determine how well each approach works in reducing the number of census block groups from 220,334 to an optimized set. The optimal number of clusters is one where the drive time computed using the census dataset and the clustered dataset shows no practical or statistically significant difference. Traditional clustering is done as a single pass using affinity propagation, *k*-means, and mean shift. Our recursive reclustering approach takes the sub-optimal clusters generated by the traditional clustering approach and continues to recluster them until the user specified stopping constraints are satisfied. The user specified constraints are described in Section 3.

	Census Block Group	Affinity Propagation	k-Means	Mean Shift
Traditional	220,334	2,113	290	43,852
Recursive Reclustering	N/A	41,442	36,666	197,675

Table 1. Number of clusters generated using traditional clustering and our recursive reclustering approach. Traditional clustering is run in a single pass, while recursive reclustering continues reclustering each cluster until the user specified constraints are satisfied.

State	**AL**	**AK**	**AZ**	**AR**	**CA**	**CO**	**CT**	**DE**	**DC**	**FL**	**GA**	**HI**	**ID**
BG	3438	534	4178	2147	23212	3532	2585	574	450	11442	5533	875	963
AP Initial	39	21	53	35	70	64	37	12	17	50	57	14	15
AP Final	704	115	872	448	4090	736	422	112	80	2150	1167	167	227
***k*-Means Initial**	9	3	5	13	2	2	2	2	2	7	2	5	4
***k*-Means Final**	588	114	651	403	3964	560	390	91	61	1910	955	152	199
MSh Initial	965	45	154	645	4039	659	1222	38	221	1470	1545	8	13
MSh Final	3061	254	3565	1875	22122	3039	2469	415	416	10567	5245	609	582

State	**IL**	**IN**	**IA**	**KS**	**KY**	**LA**	**ME**	**MD**	**MA**	**MI**	**MN**	**MS**	**MO**
BG	9691	4814	2630	2351	3285	3471	1086	3926	4985	8205	4111	2164	4506
AP Initial	72	46	38	34	40	34	27	57	54	72	67	32	40
AP Final	1809	919	555	503	644	633	222	722	848	1471	829	459	881
***k*-Means Initial**	2	12	16	3	10	4	2	3	2	4	3	10	5
***k*-Means Final**	1550	825	464	436	554	561	216	655	725	1259	723	421	747
MSh Initial	1770	1399	612	357	1337	147	375	1038	972	1313	1124	606	722
MSh Final	8897	4425	2098	1720	2813	2706	933	3628	4862	7613	3744	1742	3862

State	**MT**	**NE**	**NV**	**NH**	**NJ**	**NM**	**NY**	**NC**	**ND**	**OH**	**OK**	**OR**	**PA**
BG	842	1633	1836	922	6320	1449	15464	6155	572	9238	2965	2634	9740
AP Initial	18	34	65	24	61	20	68	57	14	55	44	37	58
AP Final	203	374	279	176	1035	307	2541	1198	164	1662	637	581	1702
***k*-Means Initial**	13	2	3	6	2	15	2	2	18	4	20	2	2
***k*-Means Final**	168	298	291	170	934	277	2289	1076	134	1376	518	460	1566
MSh Initial	60	300	11	368	1488	91	1872	3308	132	1795	839	413	2173
MSh Final	464	1359	1569	782	5934	972	14525	6041	365	8590	2604	2192	9008

State	**PR**	**RI**	**SC**	**SD**	**TN**	**TX**	**UT**	**VT**	**VA**	**WA**	**WV**	**WI**	**WY**
BG	2594	815	3059	654	4125	15811	1690	522	5332	4783	1592	4489	410
AP Initial	37	27	34	15	26	76	51	16	41	47	24	54	13
AP Final	467	117	621	177	805	2994	331	108	1028	919	280	852	99
***k*-Means Initial**	2	2	13	2	3	6	2	2	6	2	3	2	19
***k*-Means Final**	437	98	538	133	711	2804	331	95	935	780	263	732	77
MSh Initial	635	177	520	131	329	3098	178	86	1550	657	94	721	30
MSh Final	2223	710	2514	449	3689	14581	1313	336	4937	4312	1170	3565	209

Table 2. Block groups (BG), initial cluster counts using traditional clustering, and final cluster counts using reclustering with affinity propagation (AP), k-means, and mean shift (MSh) on 50 U.S. states, District of Columbia (DC), and Puerto Rico (PR).

State	AL	AK	AZ	AR	CA	CO	CT	DE	DC	FL	GA	HI	ID
AP Initial Max	39.83	157.86	39.96	42.19	52.18	33.37	12.91	14.79	1.96	42.37	35.71	26.22	71.56
AP Initial Mean	17.33	57.02	15.27	18.29	16.20	11.53	5.71	6.06	0.98	15.06	14.81	11.02	23.82
AP Final Max	5.17	27.14	2.86	6.18	1.73	3.00	2.55	2.57	0.67	2.38	3.89	2.82	5.25
AP Final Mean	2.77	11.37	1.47	3.14	0.90	1.47	1.40	1.42	0.38	1.31	2.07	1.42	2.48
***k* Initial Max**	96.62	842.15	196.60	76.32	432.38	282.96	74.63	53.34	6.95	161.22	251.95	60.81	198.29
***k* Initial Mean**	37.14	250.58	52.56	31.49	88.40	86.85	27.50	20.08	3.17	49.26	93.10	20.90	57.31
***k* Final Max**	5.47	20.22	3.65	5.99	1.64	3.90	2.48	2.92	0.74	2.47	4.35	2.75	5.59
***k* Final Mean**	3.51	13.16	2.19	3.74	1.02	2.28	1.56	1.75	0.46	1.56	2.74	1.66	3.40
MSh Initial Max	0.30	30.99	6.49	0.45	0.13	1.09	0.04	2.27	0.02	0.34	0.16	32.90	44.39
MSh Initial Mean	0.13	15.00	3.14	0.19	0.06	0.57	0.02	0.94	0.01	0.15	0.08	12.36	17.87
MSh Final Max	0.10	4.94	0.19	0.14	0.02	0.15	0.01	0.16	0.01	0.03	0.04	0.29	0.65
MSh Final Mean	0.06	2.56	0.12	0.09	0.01	0.09	0.01	0.10	0.00	0.02	0.03	0.19	0.39

State	IL	IN	IA	KS	KY	LA	ME	MD	MA	MI	MN	MS	MO
AP Initial Max	31.45	32.44	44.08	53.30	36.65	45.02	34.03	15.46	14.32	37.99	38.22	44.56	49.22
AP Initial Mean	13.00	13.94	19.54	22.40	16.40	16.20	15.48	6.62	5.98	14.36	16.48	18.69	20.86
AP Final Max	2.82	3.87	5.88	5.47	5.01	4.38	6.99	2.37	2.01	3.62	5.10	6.26	4.75
AP Final Mean	1.41	1.98	2.83	2.51	2.72	2.26	3.70	1.28	1.08	1.90	2.56	3.31	2.47
***k* Initial Max**	254.97	69.59	71.06	210.41	76.26	133.40	208.90	113.30	115.57	210.21	256.04	77.09	162.90
***k* Initial Mean**	84.21	27.39	30.71	84.97	33.39	52.14	73.44	40.78	36.48	72.79	82.67	36.70	55.42
***k* Final Max**	2.80	3.78	6.31	5.33	5.29	4.41	6.12	2.28	2.14	3.90	5.14	5.96	5.17
***k* Final Mean**	1.73	2.37	3.81	3.32	3.40	2.74	3.99	1.44	1.29	2.42	3.28	3.82	3.21
MSh Initial Max	0.20	0.18	0.54	1.20	0.11	2.54	0.51	0.13	0.11	0.33	0.30	0.49	0.64
MSh Initial Mean	0.08	0.08	0.30	0.53	0.05	1.30	0.23	0.06	0.06	0.14	0.12	0.27	0.24
MSh Final Max	0.06	0.05	0.27	0.33	0.10	0.21	0.12	0.03	0.01	0.04	0.10	0.30	0.16
MSh Final Mean	0.04	0.03	0.18	0.20	0.06	0.14	0.08	0.02	0.01	0.03	0.06	0.20	0.10

State	MT	NE	NV	NH	NJ	NM	NY	NC	ND	OH	OK	OR	PA
AP Initial Max	97.59	47.43	9.48	21.25	13.25	75.18	33.43	35.65	83.41	33.36	41.41	44.32	33.32
AP Initial Mean	33.70	19.24	3.77	9.22	5.52	22.55	12.24	14.45	30.55	13.51	16.16	13.69	12.97
AP Final Max	8.63	5.39	2.64	4.58	1.73	4.29	2.17	4.28	8.61	3.01	4.69	3.11	3.08
AP Final Mean	3.74	2.63	1.27	2.45	0.93	2.11	1.14	2.35	4.22	1.59	2.32	1.49	1.61
***k* Initial Max**	116.31	264.96	262.96	54.46	91.25	95.17	389.41	280.25	68.91	148.54	65.81	335.22	220.78
***k* Initial Mean**	39.30	108.05	51.87	22.25	33.46	30.02	84.08	89.25	27.05	49.94	27.98	113.14	74.56
***k* Final Max**	9.79	6.29	3.37	4.12	1.70	4.57	2.15	4.26	9.75	3.32	5.14	4.15	2.93
***k* Final Mean**	5.87	3.98	2.16	2.58	1.04	2.84	1.34	2.75	6.80	2.05	3.22	2.42	1.84
MSh Initial Max	11.84	1.57	65.72	0.30	0.07	6.46	0.22	0.04	2.59	0.15	0.35	1.80	0.12
MSh Initial Mean	5.65	0.79	25.43	0.15	0.03	2.96	0.08	0.02	1.25	0.07	0.12	0.97	0.04
MSh Final Max	1.60	0.33	0.13	0.12	0.01	0.46	0.03	0.01	1.38	0.02	0.12	0.20	0.03
MSh Final Mean	0.97	0.20	0.08	0.07	0.00	0.28	0.02	0.00	0.86	0.01	0.08	0.13	0.02

State	PR	RI	SC	SD	TN	TX	UT	VT	VA	WA	WV	WI	WY
AP Initial Max	11.66	6.83	35.47	76.99	49.62	65.64	22.56	27.08	38.32	39.51	37.38	39.34	71.77
AP Initial Mean	5.09	3.27	14.00	30.61	20.33	23.47	8.76	12.86	15.09	14.78	16.41	15.89	23.71
AP Final Max	1.95	2.29	4.42	8.50	4.46	3.22	3.53	7.14	3.63	3.04	6.63	4.68	5.98
AP Final Mean	1.03	1.26	2.39	4.26	2.40	1.61	1.73	3.72	1.96	1.55	3.46	2.41	2.37
***k* Initial Max**	35.34	58.78	263.30	169.23	298.00	311.28	94.10	127.22	211.39	135.02	274.62	48.80	95.73
***k* Initial Mean**	13.28	22.93	105.88	70.18	86.66	72.75	42.12	39.29	90.46	53.16	96.11	17.97	25.25
***k* Final Max**	2.41	4.46	10.50	4.45	3.14	2.96	6.73	3.55	3.35	6.13	4.88	8.42	1.82
***k* Final Mean**	1.45	2.80	6.70	2.86	1.96	1.94	4.29	2.27	2.06	3.95	3.06	4.50	1.10
MSh Initial Max	0.29	0.47	2.99	1.47	0.27	2.97	1.50	0.20	0.64	3.46	0.57	17.78	0.11
MSh Initial Mean	0.15	0.28	0.95	0.62	0.12	1.73	0.84	0.10	0.21	1.59	0.28	11.30	0.06
MSh Final Max	0.07	0.16	1.08	0.08	0.05	0.31	0.59	0.06	0.10	0.27	0.19	1.13	0.04
MSh Final Mean	0.05	0.11	0.70	0.05	0.03	0.20	0.37	0.04	0.06	0.18	0.12	0.64	0.03

Table 3. Maximum and mean values of distance between the centroid and the cluster points for initial clusters using traditional clustering and final clusters using reclustering with affinity propagation (AP), *k*-means, and mean shift (MSh) on 50 U.S. states, District of Columbia (DC), and Puerto Rico (PR).

5.1 Traditional Clustering Results

As shown in Table 1, affinity propagation reduces the census block groups from 220,334 block groups to 2,113 clusters. In Table 2, we show that affinity propagation generates a larger number of clusters in states with a larger number of block groups. We find a strong positive correlation of 0.73 between the number of census block groups and affinity clusters. The clusters generated by the traditional affinity propagation algorithm are sub-optimal, with large distances between cluster centroids and cluster points. For example, as shown in Table 3 for a sparsely populated state, such as Alaska, the average cluster centroid to cluster points is 57.0 miles with a maximum distance of 157.9 miles. We observe large distances for densely populated states, such as Rhode Island, with an average centroid to cluster point distance of 3.3 miles and a maximum distance of 6.8 miles. When considering the average across all states, the average maximum distance from the cluster centroid is 41.1 miles, and the average cluster centroid to cluster points distance is 15.9 miles.

From Table 1, we see that k-means reduces the census block groups 290 clusters. From Table 2, we see that k-means generates between 2 and 20 clusters with a median of 3 clusters. The clusters generated by the traditional k-means algorithm are highly sub-optimal, with an average distance of 58 miles between the cluster centroid and cluster points. As shown in Table 3, for Alaska we find the average cluster centroid to cluster points distance is 250.6 miles with a maximum distance of 842.1 miles. Even in densely populated states, such as Rhode Island, we find a large average cluster centroid to cluster points distance of 22.9 miles and a maximum of 58.8 miles. When considering the average across all states, the average maximum distance from the cluster centroid is 177.1 miles, and the average cluster centroid to cluster points distance is 58.0 miles.

As shown in Table 1, mean shift reduces the census block groups to 43,852 clusters. In Table 2, we observe that mean shift generates between 8 and 4,039 clusters, with a strong positive correlation of 0.87 between census block group count and cluster count. One of the challenges of mean shift is the generation of a large number of single point clusters due to the points being located in areas of low density.

We find that affinity propagation is the ideal initial clustering algorithm, as unlike k-means the user does not need to specify the number of clusters, and unlike mean shift it does not generate a large number of clusters of size 1. As shown in Figure 2, affinity propagation is more adaptive to population distribution within a state. South Dakota and Nebraska both have a land area close to 77,000 square miles, however Nebraska is a more densely populated state with over $2\times$ the population of South Dakota. While k-means generates 2 clusters for both states, affinity propagation generates 34 clusters for Nebraska, the more densely populated state, and 15 clusters for South Dakota, the more sparsely populated state. We further illustrate this in Figure 3, where we show the differences in clustering using affinity propagation, k-means and mean shift in densely populated states such as Rhode Island, California, and Illinois, and sparsely populated states such as Nevada, North Dakota, and Wyoming.

5.2 Recursive Clustering Results

As shown in Table 1, our recursive reclustering approach using affinity propagation reduces the census dataset from 220,334 block groups to 41,442 optimized clusters by

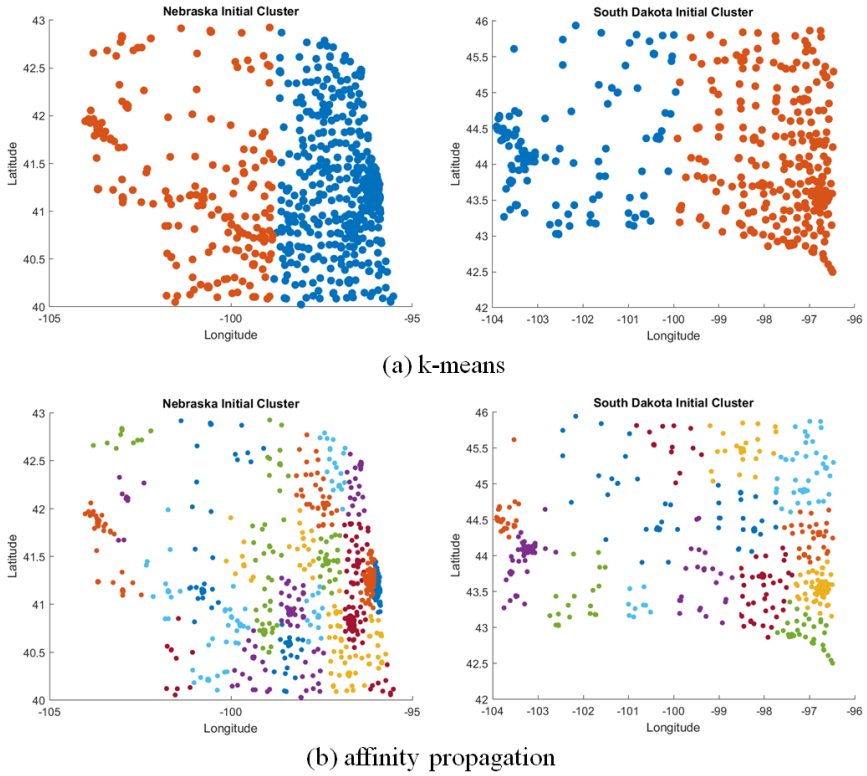


Fig. 2. Comparison of clustered census block groups for South Dakota and Nebraska. Both states have a land area of 77,000 miles, but South Dakota has a population of 869,666 while Nebraska has a population of 1,920,076. (a) *k*-means generates 2 clusters for both states and is unable to factor in the distribution of population. (b) Affinity propagation generates 34 clusters for the more densely populated state of Nebraska and 15 clusters for South Dakota. [Figure best viewed in color.]

reclustering the initial 2,113 clusters. Our approach provides a 81.2% reduction in the size of the dataset, with the highest reduction of 85.6% in Rhode Island where the 815 block groups are reduced to 117 clusters. The lowest reduction in the number of census block groups is in North Dakota, with 572 block groups being reduced to 164 clusters, i.e. a reduction of 71.3%. As shown in Table 3, the average maximum distance from the cluster centroid for all states is 4.6 miles, and the average cluster centroid to cluster points distance is 2.3 miles. For a sparsely populated state, such as Alaska the average distance from the cluster centroid to the cluster points is 11.4 miles with an average maximum of 27.1 miles. In a densely populated state, such as the District of Columbia, the average maximum distance from the cluster centroid is 0.7 miles with the average distance from the cluster centroid to all points being 0.4 miles. In densely populated

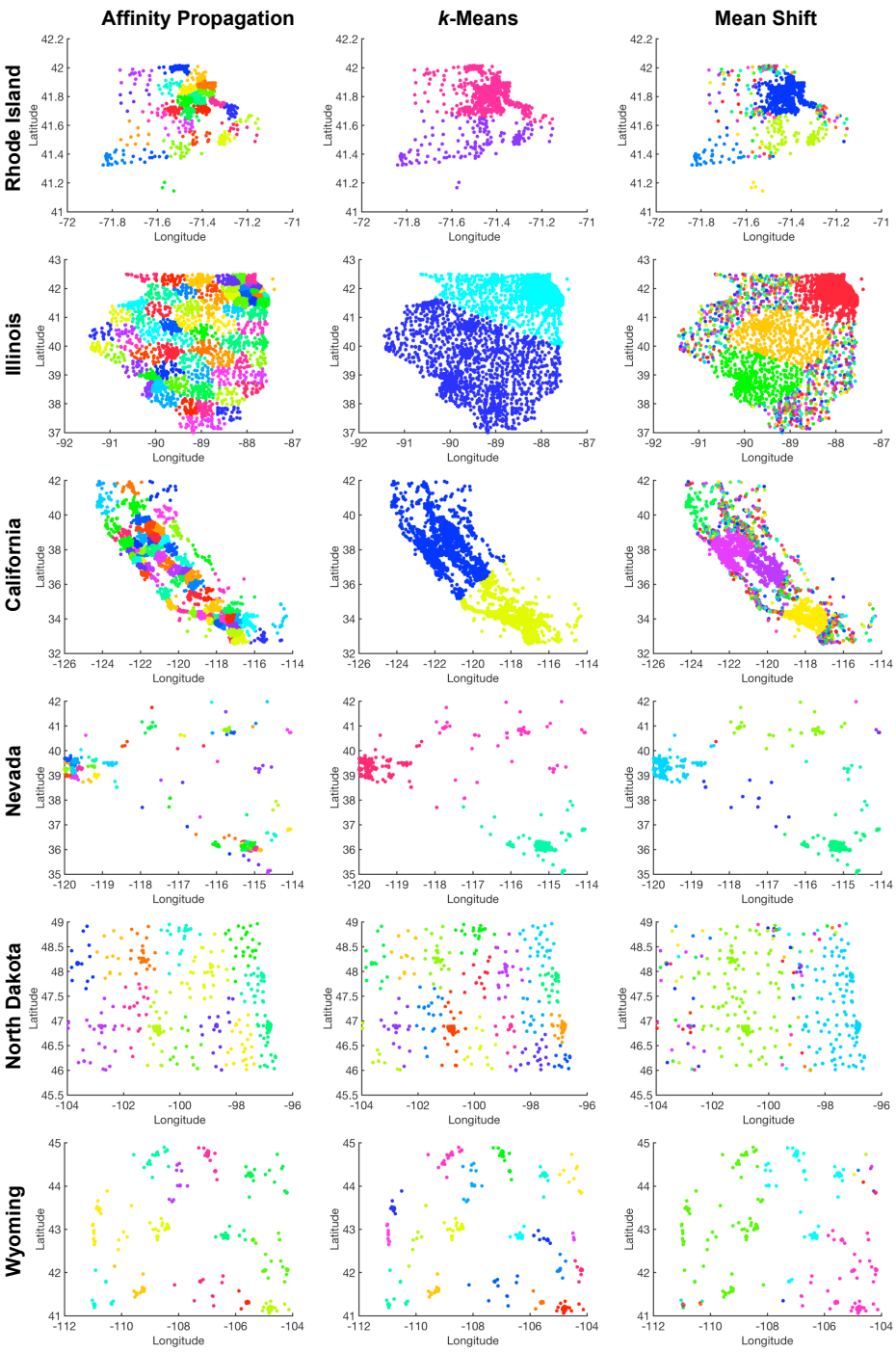


Fig. 3. Data clustered without recursion using affinity propagation, *k*-means, and mean shift. [Figure best viewed in color.]

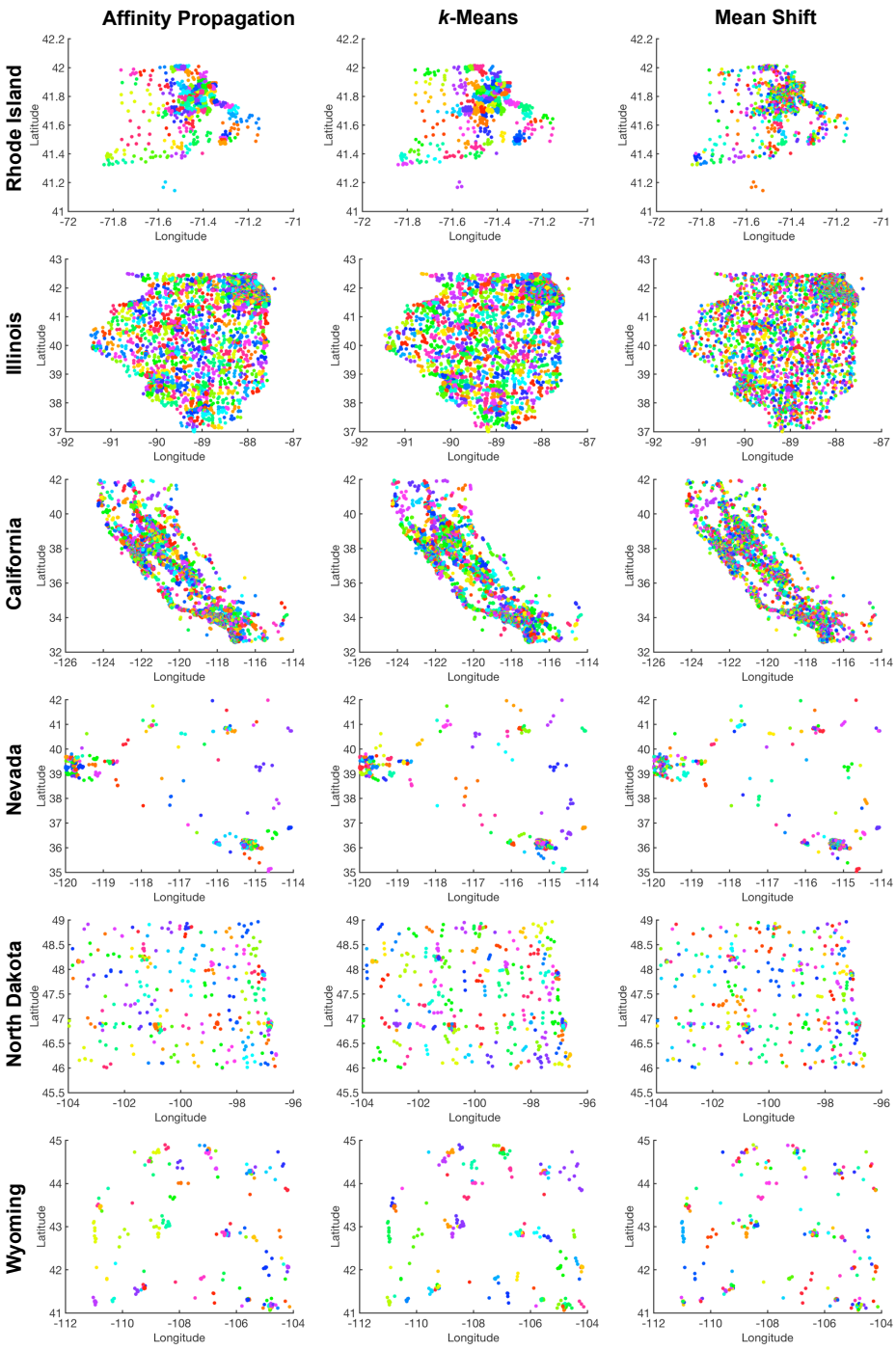


Fig. 4. Data clustered by recursive reclustering using affinity propagation, *k*-means, and mean shift. [Figure best viewed in color.]

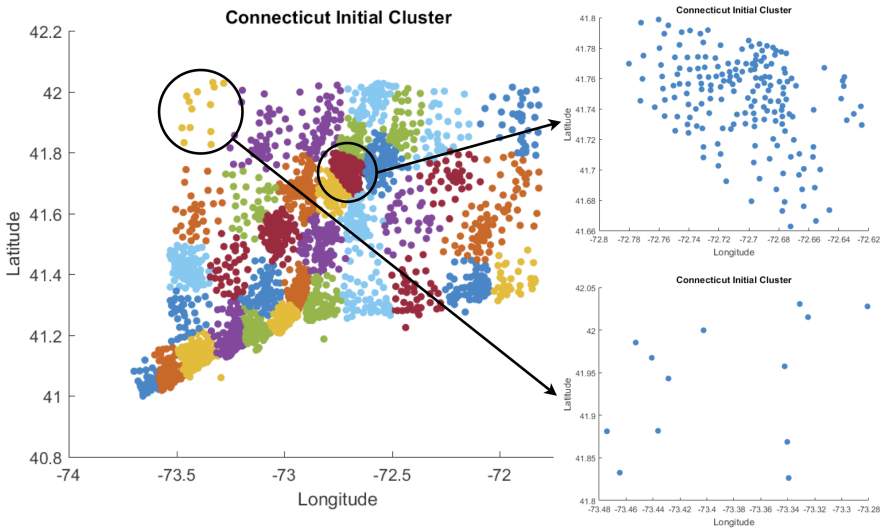


Fig. 5. A densely populated area contains several block groups in close proximity, while a sparsely populated area has larger distances between block groups. In our approach the densely populated area shown in the top right is reclustered further into smaller clusters to ensure each cluster point is less than 5 miles from the cluster centroid and the total population of the cluster is below 20,000. The sparsely populated area shown on the bottom right will also be reclustered using our approach, however our algorithm generates fewer sub clusters. [Figure best viewed in color.] [18]

states, such as the District of Columbia, small distances can have longer drive times, hence a low cluster centroid to cluster point distance is ideal.

In Table 1, we show that our recursive reclustering approach using k-means reclusters the 290 sub-optimal clusters into 36,666 optimal clusters. This results in a 83.4% reduction in the census dataset when compared to the 220,334 census block groups. The highest reduction is obtained in Rhode Island, with our approach reducing the 815 original block groups to 98 clusters, i.e. an 88.0% reduction. We see the lowest reduction in North Dakota, with the original 572 block groups being reduced to 134 clusters, i.e. a reduction of 76.6%. As shown in Table 3, the average maximum distance from the cluster centroid for all states is 4.7 miles, and the average cluster centroid to cluster points distance is 3.0 miles. Similar to our recursively reclustered affinity propagation results, we see the highest average maximum distance of 20.2 miles and highest average cluster centroid to cluster points distance of 13.2 miles in Alaska. We see the lowest average maximum distance of 0.7 miles and lowest average cluster centroid to cluster points distance of 0.4 miles in the District of Columbia. We find a strong positive correlation of 0.997 for the number of clusters generated by our recursively reclustered affinity propagation and k-means algorithms. While k-means generates the fewest number of clusters, it requires the user to provide the initial number of clusters and as a result we find affinity propagation to be the better algorithm.

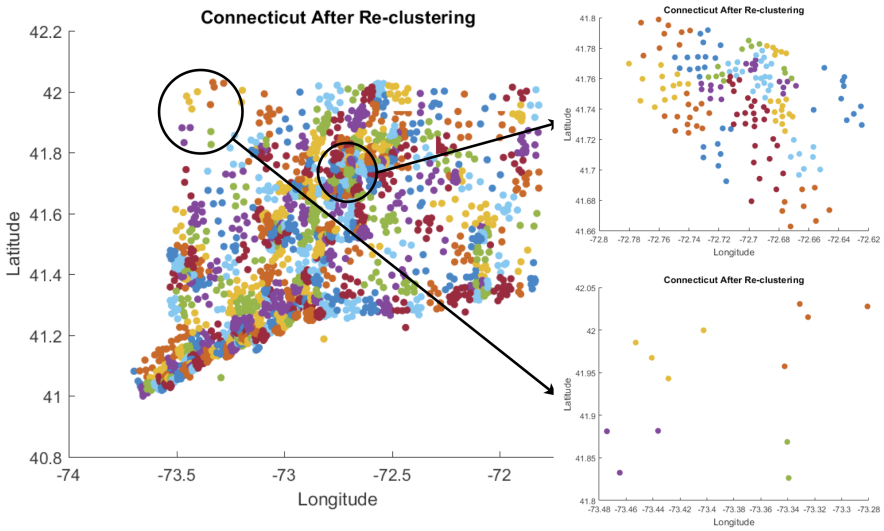
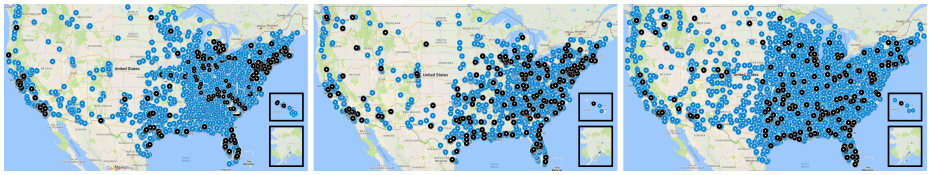


Fig. 6. A densely populated area contains several block groups in close proximity, while a sparsely populated area has larger distances between block groups. By reclustering the densely populated area into multiple smalller clusters, we ensure that the drive time differences between the raw census data and clustered data are minimized. [Figure best viewed in color.] [18]

From Table 1, we show that our recursive reclustering approach using mean shift reclusters the 43,582 sub-optimal clusters into 197,675 clusters which results in a 10.2% reduction in the census block group dataset. As shown in Figure 3 and 4, our approach does not recombine single point clusters, as a result single point clusters generated by the traditional mean shift algorithm is left as-is. From Table 3, we note that recursively reclustered mean shift shows the lowest distances for both the average cluster centroid to cluster points and maximum cluster centroid to cluster point distance. However, this distance is biased by the presence of a large number of single point clusters where the cluster centroid to cluster point distance and maximum cluster centroid to cluster point distance is 0.

5.3 Differences in Block Group Reduction

For all three recursive reclustering approaches, the variances in census block group reduction are due to the differences in land area and population distribution. From Table 2, we note that the highest percentage reduction in block groups is in Rhode Island. As shown in Figure 3 and 4, Rhode Island is a densely populated state with 1,021 individuals per square mile. Thus, fewer clustered block groups can be used to represent the state. States with low population density, such as Wyoming with 6 individuals per square mile show the lowest reduction. Finally, states such as Nevada where the population is concentrated in a few regions, we also observe a larger percentage reduction in the number of census block groups.



CVS Pharmacy | Lowe's Home Improvement | Walmart

Fig. 7. Geographic positions of all locations (blue) and 200 random locations chosen (black) for CVS Pharmacy, Lowe's Home Improvement, and Walmart. The sampled random locations represent the weighting of the original locations across the U.S.

In Figures 5 and 6, we illustrate how our recursive reclustering approach handles localities with different population densities in the state of Connecticut using affinity propagation. We show a densely populated area, Hartford, and a sparsely populated area, Salisbury, in Figures 5 and 6. As shown in Figure 5, a densely populated area has numerous block groups in close proximity, while a sparsely populated area has larger distances between block groups. Thus, as shown in Figure 6, our approach reclusters the densely populated area into a higher number of clusters and the sparsely populated into a smaller number of clusters based on the constraints described in Section 3.

6 Evaluation

In order to evaluate the effectiveness of our approach, we sample 200 random locations each from three major retailers in the United States—CVS Pharmacy, Lowe's Home Improvement, and Walmart—which each have 9,807, 1,741, and 5,746 locations respectively. We obtain a total of 600 locations. As shown in Figure 7, the random locations represent the weighting of the original distribution of locations across the country.

A typical location selection procedure evaluates the effectiveness of a location by placing a trade area at a given radius around a location and computing the average drive time to the intended location for each potential customer within the trade area. We use the Google Maps Distance Matrix API [17] to compute the drive times for the census block group, traditional clustered, and recursively reclustered data to existing business locations using the approach discussed in Section 4. Since our approach in Section 4 yields a bounding box as opposed to a circle, we use a half-size for the bounding box of 5 miles instead of a radius.

6.1 Comparison to Traditional Clustering

We use a paired t-test to determine if the differences in drive times for census block group and traditionally clustered data are significantly different by testing the following hypotheses:

NULL: the mean drive time for census block group data is no different from the mean drive time for traditionally clustered data.

Alternate: the mean drive time for census block group data is different from the mean drive time for traditionally clustered data.

			Traditional			Recursively Reclustered		
Dataset	Actual	Census	AP	k-Means	Mean Shift	AP	k-Means	Mean Shift
CVS	9807	9807	5744	715	2622	9788	9799	9802
Lowe's	1741	1741	972	99	499	1741	1741	1741
Walmart	5746	5743	2793	341	1640	5723	5725	5735

Table 4. Comparison of the number of actual retail store locations, and the number of retail stores with at least one census block group or clustered block group within 5 miles.

	Census vs. Traditionally Clustered			Census vs. Recursively Reclustered		
	AP	k-Means	Mean Shift	AP	k-Means	Mean Shift
CVS	Retain$*$	Retain$*$	Retain$*$	Retain	Retain	Reject
Lowe's	Retain$*$	Retain$*$	Retain$*$	Retain	Reject	Reject
Walmart	Retain$*$	Retain$*$	Retain$*$	Retain	Retain	Reject

Table 5. Hypothesis test results to determine if differences in drive times are statistically significant. Retain indicates that we failed to reject the NULL hypothesis, indicating that there is no difference in drive time. Reject indicates that we reject the NULL hypothesis, indicating that there is a difference in drive time. A $*$ indicates that results were computed with less than 200 locations due to a cluster centroid being further than 5 miles away.

From Table 4, we observe that all CVS, Lowe's, and Walmart locations contain at least one census block group within 5 miles. We note that traditional clustering algorithms are ill suited as over 40% of the locations no longer have a clustered block group within 5 miles when using affinity propagation, 90% when using k-means, and over 70% when using mean shift.

From Table 5, we see no statistically significant differences in drive time for all three datasets using all three algorithms. However, 91 CVS, 93 Lowe's, and 95 Walmart locations have no clustered block groups within 5 miles when using affinity propagation. When using k-means 190 CVS, 192 Lowe's, and 187 Walmart locations have no clustered block groups within 5 miles. Finally, for mean shift 157 CVS, 137 Lowe's and 153 Walmart locations do not have a clustered block group within 5 miles. From Table 6, we observe that the 95% confidence interval of the differences are between 20 seconds to 327 seconds. Traditional clustering is unsuitable to generate optimal clusters for improving drive time computations, as over 40% of the locations in our sampled dataset have a cluster centroid that is more than 5 miles away.

6.2 Comparison to Recursive Reclustering

We use a paired t-test to determine if the differences in drive times for census block group and recursively reclustered data are significantly different by testing the following hypotheses:

NULL: the mean drive time for census block group data is no different from the mean drive time for recursively reclustered data.

	Census vs. Traditionally Clustered			Census vs. Recursively Reclustered		
	AP	*k*-Means	Mean Shift	AP	*k*-Means	Mean Shift
CVS	[-39.9, 45.5]∗	[-104.1, 108.9]∗	[-38.0, 24.4]∗	[-8.0, 12.3]	[-15.8, 13.0]	[10.1, 20.2]
Lowe's	[-45.5, 34.8]∗	[-327.2, 239.6]∗	[-22.2, 19.2]∗	[-12.6, 6.0]	[8.6, 28.3]	[11.6, 20.3]
Walmart	[-61.5, 21.9]∗	[-271.0, 80.3]∗	[-137.3, 41.4]∗	[-5.3, 13.6]	[-0.3, 26.9]	[4.0, 20.2]

Table 6. 95% confidence interval of difference in drive times. The values for the confidence interval are provided in seconds. A $*$ indicates that results were computed with less than 200 locations due to a cluster centroid being further than 5 miles away.

Alternate: the mean drive time for census block group data is different from the mean drive time for recursively reclustered data.

From Table 4, we observe that all CVS, Lowe's, and Walmart locations contain at least one census block group within 5 miles. Our recursively reclustering approach shows 0.4% data loss, with at most 19 out of 9,807 total CVS location and 23 out of 5,746 total Walmart locations having no recursively reclustered block group within 5 miles.

From Table 5, we see no statistically significant differences in drive time for all three datasets using affinity propagation. We see no statistically significant difference in drive times for Walmart and CVS for *k*-means, while a significant difference in drive time was observed for Lowe's, the difference in drive times is under 30 seconds. While we see a statistically significant difference in drive time for mean shift, the difference is less than 30 seconds and thus practically insignificant.

From Table 6, we observe that our recursive reclustering approach yields drive times within 30 seconds of the actual drive time obtained from the census block group data. For the 200 random CVS Pharmacy locations, we computed drive times to 34,888 locations when using the census data, but only 5,516 when using affinity propagation, 4,990 using *k*-means, and 32,593 using mean shift. For the 200 random Lowe's Home Improvement locations we compute 21,249 drive times using the census data, but only 3,604 using affinity propagation, 3,202 using *k*-means, and 19,105 using mean shift. Finally, for the 200 random Walmart locations we computed drive times to 17,876 locations using the census data, and only 2,997 locations using affinity propagation, 2,710 locations using *k*-means, and 16,528 locations using mean shift.

Our recursively reclustered affinity propagation and *k*-means approach provides a $6\times$ reduction in the number of computations with no practically perceivable difference in the drive times. For example, as shown in Figure 8 for a random location denoted by the diamond symbol and located at coordinates (41.766458,-72.677643), we generate 253 potential customers groups in a 5 mile bounding box using the census block group data with an average drive time of 10 minutes 14 seconds. Our approach generates 33 clustered customer groups with an average drive time of 10 minutes 5 seconds [18].

6.3 Differences in Actual and Recursively Reclustered Drive Times

Drive time differences between the census and recursively reclustered data arise due to the number of points within the trade area for a given location. In Figure 9, we observe

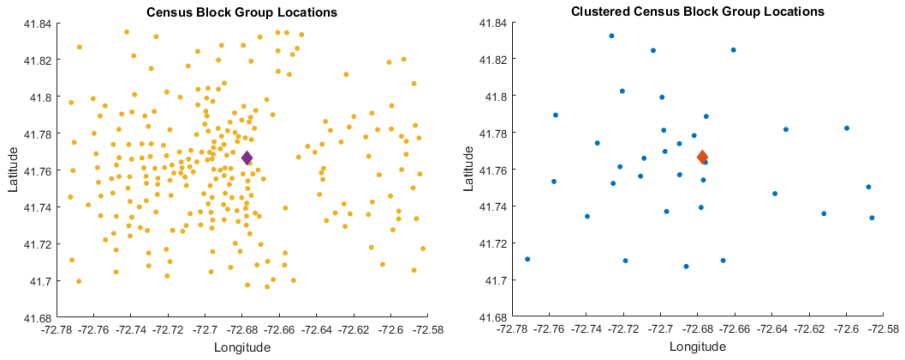


Fig. 8. Effect of clustering on reducing the number of drive time computations in a urban location, such as Hartford, CT. The diamond indicates a proposed location, and the circles indicate block groups. The figure on the left shows the raw census block group data, while the figure on the right shows the clustered block group data [18].

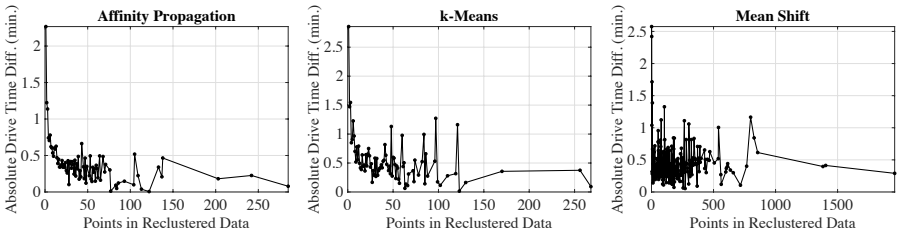


Fig. 9. Drive time differences measured in minutes for census vs. clustered census data using affinity propagation, *k*-means, and mean shift. Drive time differences reduce as the number of clustered points in the neighborhood of a proposed location increases.

that for all three clustering algorithms, the differences between actual drive times and recursively reclustered drive times increase as the number of block group clusters decreases. In sparsely populated areas, census block groups are separated by larger spatial distances. In such areas, we observe drive time differences up to 2 minutes. In densely populated areas, where census block groups are spatially closer we observe lower drive time differences, many of which are within 30 seconds. Residents in sparsely populated areas are more likely to be accepting of slightly longer differences in drive times due to the inherent sparsity of resources.

7 Discussion

In this work, we perform a comparison of our approach on constrained recursive reclustering introduced in our prior work [18] for affinity propagation, on two additional clustering algorithms, i.e., *k*-means and mean shift, and on a variety of site location datasets. Unlike the affinity propagation algorithm used in our original work, both *k*-means and mean shift require the pre-specification of user-defined parameters, such

as cluster count for k-means and kernel bandwidth for mean shift. We use the approach of silhouette score computation to select an optimal cluster count in k-means, while we use the 30^{th} percentile of pairwise distances as the kernel bandwidth in mean shift. Our recursive reclustering approach provides reductions of 81.2%, 83.4%, and 10.2% for affinity propagation, k-means, and mean shift respectively when compared to the 220,334 census block groups. Using 200 randomly sampled locations each from Lowe's, CVS, and Walmart, we show that compared to the original block groups there is no statistically significant difference in drive time computations when using clusters generated by constrained recursive reclustering with affinity propagation for any of the three businesses, and with k-means for CVS and Walmart. While statistically significant differences are obtained with k-means for Lowe's and with mean shift for all three businesses, the differences are negligible, with the mean difference within each location set being within 30 seconds.

While we do not use the haversine distance to compute the drive time itself, we do use it as a heuristic in the constraint set in Subsection 3.3 to perform cluster splitting. In future, we will perform statistical comparison on using the haversine distance heuristic and on using higher resolution geographical data such as locations of natural relief, traffic patterns, and differences in speed limits of local roads to perform cluster splitting. In this work, we use pre-defined thresholds of 10 for the cluster point counts, 20,000 individuals for the population cutoff, and 5 miles for the distance bound. In future, we will perform statistical analyses with a range of population and distance thresholds. Currently, our approach performs splitting of clusters till they satisfy the constraint set. However, in the event that small clusters are created during the initial clustering, as in the mean shift approach, it is possible for the small clusters to be re-grouped into a larger cluster that still satisfies the constraint set. In future work, we will perform bottom-up cluster regrouping to obtain the minimal number of clusters that just meet the constraint set. Future work will also include a comparison of computation performance and statistical analysis on drive time computation results for constrained recursive reclustering on other groups of clustering algorithms such as spectral and hierarchical clustering.

References

1. Aras, H., Erdoğmuş, Ş., Koç, E.: Multi-criteria selection for a wind observation station location using analytic hierarchy process. Renewable Energy **29**(8), 1383–1392 (2004)
2. Athas, W.F., Adams-Cameron, M., Hunt, W.C., Amir-Fazli, A., Key, C.R.: Travel distance to radiation therapy and receipt of radiotherapy following breast-conserving surgery. JNCI **92**(3), 269–271 (2000)
3. Banaei-Kashani, F., Ghaemi, P., Wilson, J.P.: Maximal reverse skyline query. In: Proc. ACM SIGSPATIAL. pp. 421–424 (2014)
4. Blanchard, T., Lyson, T.: Access to low cost groceries in nonmetropolitan counties: Large retailers and the creation of food deserts. In: Measuring Rural Diversity Conference Proceedings, November. pp. 21–22 (2002)
5. Bradley, P., Bennett, K., Demiriz, A.: Constrained k-means clustering. Microsoft Research, Redmond pp. 1–8 (2000)
6. Branas, C.C., MacKenzie, E.J., Williams, J.C., Schwab, C.W., Teter, H.M., Flanigan, M.C., Blatt, A.J., ReVelle, C.S.: Access to trauma centers in the united states. JAMA **293**(21), 2626–2633 (2005)

7. Carr, B.G., Branas, C.C., Metlay, J.P., Sullivan, A.F., Camargo, C.A.: Access to emergency care in the united states. Annals of emergency medicine **54**(2), 261–269 (2009)
8. Çebi, F., Otay, I.: Multi-criteria and multi-stage facility location selection under interval type-2 fuzzy environment: a case study for a cement factory. IJCIS **8**(2), 330–344 (2015)
9. Census, U.: 2010 us census block group data. http://www2.census.gov/geo/docs/reference/cenpop2010/blkgrp/CenPop2010_Mean_BG.txt (2010)
10. Chen, L., Zhang, D., Pan, G., Ma, X., Yang, D., Kushlev, K., Zhang, W., Li, S.: Bike sharing station placement leveraging heterogeneous urban open data. In: Proc. ACM Ubicomp. pp. 571–575 (2015)
11. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence **24**(5), 603–619 (2002)
12. Farber, S., Morang, M.Z., Widener, M.J.: Temporal variability in transit-based accessibility to supermarkets. Applied Geography **53**, 149–159 (2014)
13. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007)
14. Ghaemi, P., Shahabi, K., Wilson, J.P., Banaei-Kashani, F.: Optimal network location queries. In: Proc. ACM SIGSPATIAL. pp. 478–481 (2010)
15. Ghaemi, P., Shahabi, K., Wilson, J.P., Banaei-Kashani, F.: Continuous maximal reverse nearest neighbor query on spatial networks. In: Proc. ACM SIGSPATIAL. pp. 61–70 (2012)
16. Goodman, D.C., Fisher, E., Stukel, T.A., Chang, C.h.: The distance to community medical care and the likelihood of hospitalization: is closer always better? Am. J. Public Health **87**(7), 1144–1150 (1997)
17. Google: Google Maps Distance Matrix API. https://developers.google.com/maps/documentation/distance-matrix/ (2017)
18. Gwinn, D., Helmick, J., Banerjee, N.K., Banerjee, S.: Optimal estimation of census block group clusters to improve the computational efficiency of drive time calculations. In: GIS-TAM. pp. 96–106 (2018)
19. Jiao, J., Moudon, A.V., Ulmer, J., Hurvitz, P.M., Drewnowski, A.: How to identify food deserts: measuring physical and economic access to supermarkets in king county, washington. Am. J. Public Health **102**(10), e32–e39 (2012)
20. Kahraman, C., Ruan, D., Doan, I.: Fuzzy group decision-making for facility location selection. Information Sciences **157**, 135–153 (2003)
21. Karamshuk, D., Noulas, A., Scellato, S., Nicosia, V., Mascolo, C.: Geo-spotting: mining online location-based services for optimal retail store placement. In: Proc. ACM SIGKDD. pp. 793–801 (2013)
22. Kuo, R., Chi, S., Kao, S.: A decision support system for locating convenience store through fuzzy ahp. Computers & Industrial Engineering **37**(1), 323–326 (1999)
23. Li, Y., Zheng, Y., Ji, S., Wang, W., Gong, Z., et al.: Location selection for ambulance stations: a data-driven approach. In: Proc. ACM SIGSPATIAL. p. 85 (2015)
24. Lloyd, S.: Least squares quantization in pcm. IEEE Trans. Information Theory **28**(2), 129–137 (1982)
25. Love, R.F., Morris, J.G.: Mathematical models of road travel distances. Management Science **25**(2), 130–139 (1979)
26. Nallamothu, B.K., Bates, E.R., Wang, Y., Bradley, E.H., Krumholz, H.M.: Driving times and distances to hospitals with percutaneous coronary intervention in the united states. Circulation **113**(9), 1189–1195 (2006)
27. Nattinger, A.B., Kneusel, R.T., Hoffmann, R.G., Gilligan, M.A.: Relationship of distance from a radiotherapy facility and initial breast cancer treatment. JNCI **93**(17), 1344–1346 (2001)
28. Park, H.S., Jun, C.H.: A simple and fast algorithm for k-medoids clustering. Expert systems with applications **36**(2), 3336–3341 (2009)

29. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. J. Mach. Learn. Res **12**, 2825–2830 (2011)
30. Qu, Y., Zhang, J.: Trade area analysis using user generated mobile location data. In: Proc. International Conference on World Wide Web. pp. 1053–1064. ACM (2013)
31. Rokach, L., Maimon, O.: Clustering methods. In: Data mining and knowledge discovery handbook, pp. 321–352. Springer (2005)
32. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
33. Statista: Total number of walmart stores worldwide from 2008 to 2018. https://www.statista.com/statistics/256172/total-number-of-walmart-stores-worldwide/ (2018)
34. Tzeng, G.H., Chen, Y.W.: The optimal location of airport fire stations: a fuzzy multi-objective programming and revised genetic algorithm approach. Transportation Planning and Technology **23**(1), 37–55 (1999)
35. Tzeng, G.H., Teng, M.H., Chen, J.J., Opricovic, S.: Multicriteria selection for a restaurant location in taipei. International journal of hospitality management **21**(2), 171–187 (2002)
36. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al.: Constrained k-means clustering with background knowledge. In: ICML. vol. 1, pp. 577–584 (2001)
37. Wang, F., Chen, L., Pan, W.: Where to place your next restaurant?: Optimal restaurant placement via leveraging user-generated reviews. In: Proc. ACM CIKM. pp. 2371–2376 (2016)
38. Wang, Y., Jiang, W., Liu, S., Ye, X., Wang, T.: Evaluating trade areas using social media data with a calibrated huff model. ISPRS International Journal of Geo-Information **5**(7), 112 (2016)
39. Xiao, X., Yao, B., Li, F.: Optimal location queries in road network databases. In: IEEE ICDE. pp. 804–815 (2011)
40. Xu, M., Wang, T., Wu, Z., Zhou, J., Li, J., Wu, H.: Demand driven store site selection via multiple spatial-temporal data. In: Proc. ACM SIGSPATIAL. p. 40 (2016)
41. Yang, J., Lee, H.: An ahp decision model for facility location selection. Facilities **15**(9/10), 241–254 (1997)
42. Yong, D.: Plant location selection based on fuzzy topsis. Int. J. Adv. Manuf. Technol **28**(7), 839–844 (2006)
43. Yu, Z., Tian, M., Wang, Z., Guo, B., Mei, T.: Shop-type recommendation leveraging the data from social media and location-based services. ACM TKDD **11**(1), 1 (2016)
44. Yu, Z., Zhang, D., Yang, D.: Where is the largest market: Ranking areas by popularity from location based social networks. In: IEEE UIC/ATC. pp. 157–162 (2013)