CS221 Project Final Report

Project Title: Improved Sales Territory Management Clusters
Team Members: Andrew Lawlor
Emails: adlawlor@stanford.edu

1 Introduction

Salesforce is a popular Customer Relationship Management (CRM) solution that (among other things) tracks Salesforce Automation (SFA) data including which customers are in which territory. Each territory is (typically) owned by one sales rep. It has become a painful and manual process to design a company's sales territories. Sales management must iterate through many customer-to-territory assignments to balance a variety of constraints. Chief among these constraints is geographic proximity of customers within a territory, allowing a single sales rep (ideally located within the territory) to manage customers in a given region. But this is not enough. The territories must also balance a variety of other constraints as well. This project proposes to solve this perennial Sales department challenge through the application of AI/machine learning techniques to produce geographically balanced territories for any company.

2 Literature Review

There are a number of sales territory planning tools on the market today, including TerrAlign, AlignMix, and eSpatial. Most of these tools are rules-based and cannot offer the flexibility that an AI/ML driven offering would bring. Salesforce offers the Salesforce Maps Territory Planning product to help optimize sales territories. This is a feature rich tool, but also appears to be rules based. Outside the Salesforce ecosystem there are general purpose optimization solutions including Gurobi (http://www.gurobi.com).

3 Dataset

This project will focus on the Commercial Real Estate (CRE) industry and will leverage geospatial data for customers and properties from this industry, with which I have access through my company's industry focus. Thousands of locations with latitude and longitude features will be placed into an optimized cluster. Minimal pre-processing is expected.
There are a total of 71,008 properties in the dataset, all in the lower 48 states. Each of these properties (consider them clients) is geocoded with Latitude and Longitude.

4 Baseline & Oracle

Baseline: The baseline will leverage k-means to balance territories by geospatial (lat/long) feature data only. This will produce territories of limited usefulness in the real world.

Oracle: The oracle will leverage spectral clustering on an arbitrary number of dimensions/constraints to balance on simultaneously. Metrics to track: (average distance to territory center, territory size (sq miles), customer revenue ($), number of customers (count), customer size (avg # employees), industry (count/per).

5 Main Approach

The main approach will be to apply spectral clustering to this territory optimization problem. The spectral clustering algorithm is a natural fit for this problem due to the geospatial nature of territory planning. Spectral clustering is preferred over k-means because it can find natural clusters that are not spherical in shape. The classic example is overlapping crescent shapes. Spectral clustering can find these separate territories whereas k-means will force them into separate spheres, a non-optimum solution for sales territory management.

6 Evaluation Metric

The success of a territory management optimization solution can be measured by confirming the newly assigned territories are:
1. Is each territory geographically sensible?
2. Do the territories demonstrate business balance (are workloads/revenues fair between territories).
3. Would a real-world Sales leadership team put the territory mapping into use?

7 Results & Analysis

The ML pipeline is as follows:
1. The full properties dataset is loaded
2. Extraneous data from outside the lower 48 states is filtered out.
3. NaNs are filtered out.
4. The geospatial (lat/long) data is scaled as K-means uses Euclidean distance.
5. Train/Test Split

> Territories are usually in effect for one calendar year coinciding with the company's fiscal year. New customers and prospects are added/removed from a given territory throughout the year. We will model this scenario with a train/test split of the data. The territories are formed (fit) using the training dataset (80%). The full dataset (train + test) will then be used to evaluate (predict) how well the territories fit unseen data.

5. Territories (clusters) are identified by calling the fit() method from the sklearn KMeans package. See **figure 1** for a geographical representation of the territories.

> For this project we set K=50 to represent 50 territories across the lower 48 US states. This is a fairly common standard for enterprise class sales territories.

6. The total cost (known as inertia for k-Means) is calculated for the training dataset.
7. Additional scores are calculated on the training dataset to measure the quality of the clusters including Silhouette score, Davies-Bouldin index, and Calinski-Barabasz Index.
8. Territories (clusters) are identified by calling the predict() method on the train+test dataset.

> The sklearn.kmeans.predict() method does NOT affect the clusters. Instead, it measures how the full dataset matchs the clusters identified by fit() above.

9. The total cost (known as inertia for k-Means) is calculated for the training+test dataset.
10. Additional scores are calculated on the training+test dataset to measure the quality of the clusters including Silhouette score, Davies-Bouldin index, and Calinski-Barabasz Index.

Costs and scores for K-Means:
Total Cost (Inertia) for K-means clusters (training data): 1563.33
Silhouette Score for K-means (training data): 0.4272
Davies-Bouldin Index for K-means (training data): 0.7338
Calinski-Harabasz Index for K-means (training data): 83083.7030
Total Cost (Inertia) for K-means clusters (full dataset): 1957.92
Silhouette Score for K-means (full dataset): 0.4269
Davies-Bouldin Index for K-means (full dataset): 0.7354

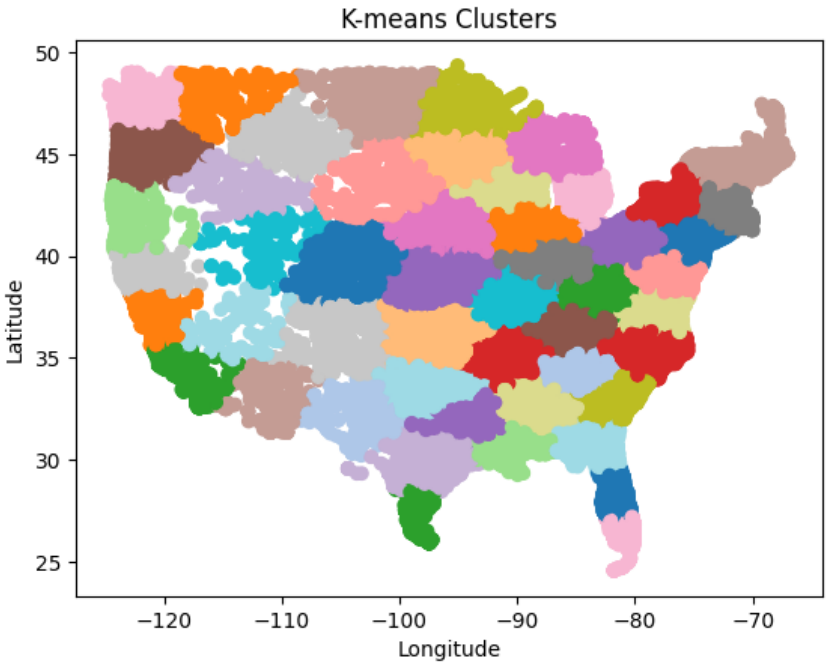

Figure 1: k-Means territories for the training dataset

Calinski-Harabasz Index for K-means (full dataset): 103613.9197

Interpretation of results:

> **Inertia**: We see the training+test cost (1,958) increased from the training cost (1,563). This is expected as the centroids were set on the training data and didn't move for the full training+test dataset.
> **Silhouette** Score (higher = better): The train and full scores are nearly identical => Excellent stability and consistency.
> **Davies-Bouldin** (lower = better): Only a small degradation => very stable.
> **Calinski-Barabasz** (higher = better): Clusters are more distinct with full data.
> Overall: *No signs of overfitting*.

Our pipeline continues.
11. Next we fit the training data using Spectral Clustering. See **figure 2** for a geographical representation of the territories.

> We use the same K=50 clusters. As this is geospatial data we us affinity=nearest_neighbors with a sparse dataset to keep computations down.

12. Silhouette, Davies-Bouldin and Calinski-Harabasz scores are calculated for Spectral clustering. (note that Inertia applies only to K-Means).

Costs and Scores for Spectral Clustering:
Silhouette Score for Spectral Clustering (training data): 0.3129
Davies-Bouldin Index for Spectral Clustering (training data): 0.7547
Calinski-Harabasz Index for Spectral Clustering (training data): 51131.5489
Silhouette Score for Spectral Clustering (full dataset): 0.3081
Davies-Bouldin Index for Spectral Clustering (full dataset): 0.7663
Calinski-Harabasz Index for Spectral Clustering (full dataset): 62599.0913

Interpretation of results:

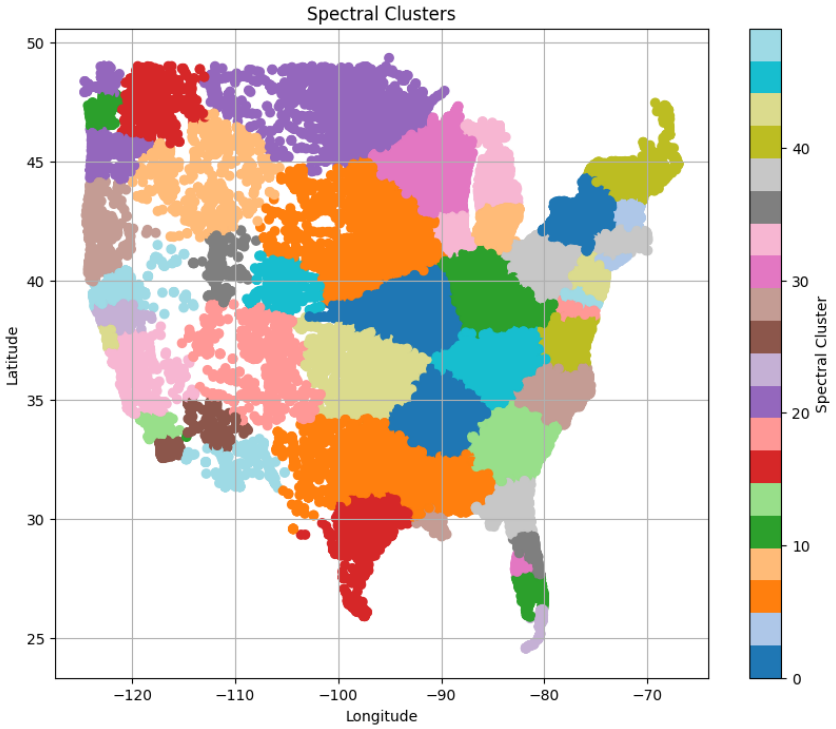

Figure 2: Spectral territories for the training dataset

Silhouette (higher = better): Slight drop from train to full, but overall lower than K-Means.
Davies-Bouldin (lower = better): Slight degradation on full dataset and also worse than K-Means.
Calinski-Harabasz (higher = better): CH increases with more data, but is still far below K-Means.
Overall: Spectral clusters are less compact (lower Silhouette) and less well-separated (higher DB index) than K-Means clusters. Performance drops slightly when applied to full dataset (as expected).

Our pipeline continues:
13. Let's look beyond the numbers and focus on the clusters side-by-side to understand what the algorithms did. See **figure 3** (no boundaries) and **figure 4** (with boundaries highlighted).

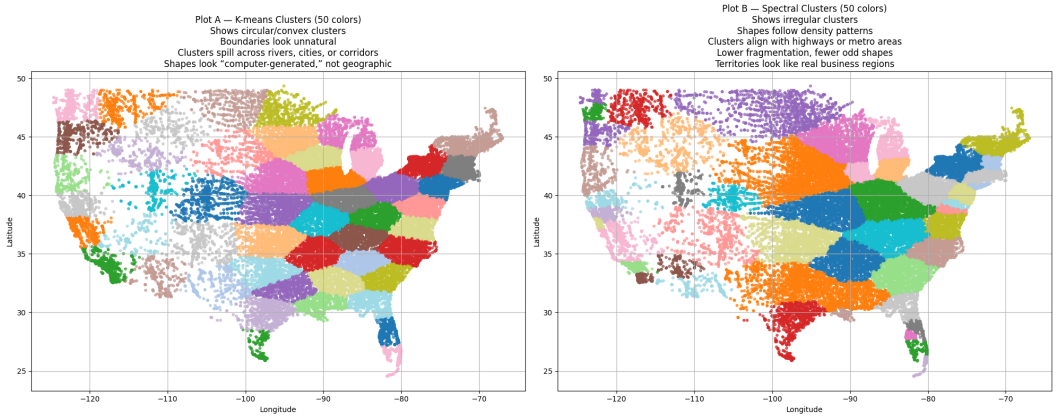

Figure 3: Side-by-Side Territories

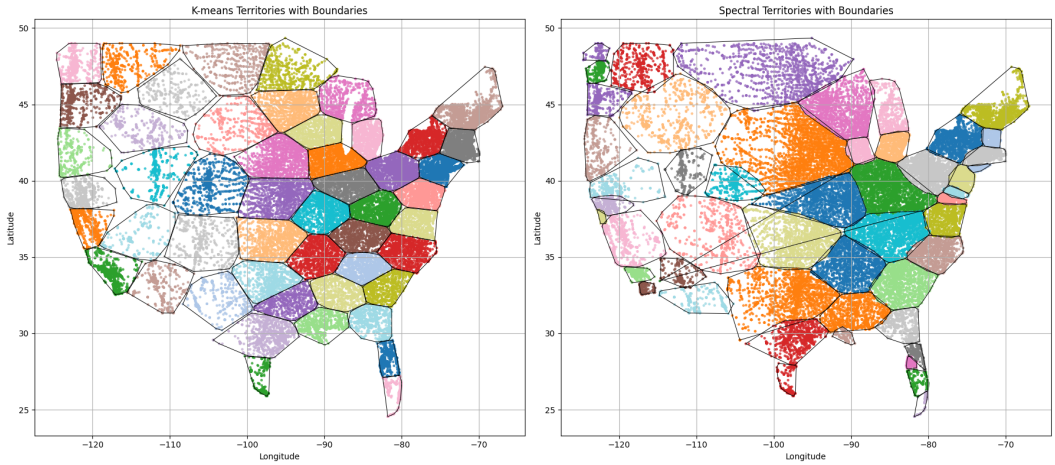

Figure 4: Side-by-Side Territories with Boundaries

14. Statistical analysis of the two clustering algorithms.

> The Standard deviation is 677 for K-means territories and 894 for Spectral clustering territories. This shows that clients (properties) are more evenly spread out with Spectral clustering. (desirable).

Interpretation of results:

> Although the territories from K-Means have better scores than from Spectral clustering, they don't work as well in the real world. K-Means territories ignores population-dense metropolitan regions, resulting in territories that are unbalanced in customer count. Conversely, the Spectral territories are sensitive to population centers and geographic features and create additional territories in those metropolitan areas. For example, Florida has 4 territories from K-Means but 6 from Spectral clustering. Similarly, population dense areas in the Bay Area, LA, Mid-Atlantic and NY/NE have additional territories. The territories generated by Spectral Clustering are more useful to a real-world business.

8 Error Analysis

The following errors were observed in this analysis:
1. **Misalignment between internal metrics and business objectives**.
Silhouette, DB and CH measure compactness and separation in Euclidean space, but real territory quality depends on other attributes including population density, geography, travel feasibility and workload balance.
2. **K-Means' failure in Metropolitan Areas**.
K-Means assumes spherical clusters, equal variance, and linear separability. These assumptions break in high-density metro areas where account distributions are non-linear and non-convex.

The standard deviation for K-Means = 677 while for Spectral clustering = 894. This indicates a 'tighter' distribution with K-Means. This again is misleading as Spectral Clustering redistributes clusters in proportion to regional density, not raw point count. Businesses want territories that reflect population/client density and natural geographic features.

9 Future Work

The future work for this project would tackle the Oracle.
Adding additional constraints to balance between territories (clusters) such as:

1. Total revenue of clients within a territory
2. Number of clients within a territory
3. Average customer size (by number of employees)
4. Industries within a territory
5. [many others]

To balance on these additional attributes turns this project into a constraint satisfaction problem (CSP). Adding such hard constraints adds exponential complexity to the problem. It would be fun to develop a solution to this territory management solution for an arbitrary number of added constraints. I believe such a solution would be commercially viable as an App on the Salesforce AppExchange (much like the Apple AppStore), a technology my company has deep experience developing.

10 Ethical Considerations

One consideration sales leaders must address when setting sales territories is equity between territories and the sales reps that sell into them. Does each territory owner (i.e. the sales rep) have a roughly equal ability to meet their sales quota based on the makeup of the clients in their territory.

Do the sales reps that own dense, urban territories have all the most lucrative clients? Do the sales reps that own more rural territories have to travel further to see their customers?

To mitigate these issues, Sales Management can consider additional constraints when planning their territories. By factoring in customer density or total size of a territory (i.e square miles) as additional constraints the resulting territory map will be more equitable. These considerations are included in the Oracle/Future Work for this project.

11 Code

The code for this project is located in a Google Colab (open sharing model):
https://colab.research.google.com/drive/1BPccVhirb9tw4CTyY9iwHtN1nOKpiNSt?usp=sharing

References

I referenced the following papers while developing this project:

1. "Constrained clustering + sales territory design" by DeSarbo, W.S., and Mahajan, V. (1984)

This problem aligns with the constrained clustering framework of DeSarbo and Mahajan (1984), who show that incorporating a priori business constraints is essential for realistic sales territory design.

2. "Spectral clustering for spatially contiguous regions" by Yuan, S. and Tan, P. (2019)

The use of spectral clustering for territory delineation is consistent with Yuan et al. (2019), who show that spatially constrained spectral methods can produce contiguous regions that balance homogeneity and size.