

# Unsupervised Classification of Oil Customers for GasDay<sup>TM</sup>

Adam Mallen

Department of Mathematics, Statistics, and Computer Science

December 16, 2010

- ▶ Oil companies use GasDay's<sup>TM</sup> software and models to predict their customers' oil use.
- ▶ This project's goal is to cluster customers to see if different 'types' of customers exist.
  - ▶ This task is *unsupervised* clustering (no classes).
- ▶ This project is *exploratory*.
  - ▶ Negative results are still useful.
  - ▶ Many decisions and parameters are chosen arbitrarily.

## Customer Data

- ▶ Customer ID
- ▶ Lat
- ▶ Long
- ▶ # of Tanks
- ▶ Size of Tanks
- ▶ Air or Water Heating
- ▶ k-Factor

## Delivery Data

- ▶ Customer ID
- ▶ Date
- ▶ Delivery Amount

## Weather Data

- ▶ Date
- ▶ Hour
- ▶ Temperature

- ▶ All customer data fields were used as features, except ...
  - ▶ Oil Capacity = (# of tanks) \* (size of tanks) was used instead of just the size of the tanks
- ▶ To use the delivery time series a linear model was fit and the parameters  $\beta_0$  and  $\beta_1$  were used as features.

# Delivery Time Series Model

Let ...

- ▶  $D_k$  be the amount of oil in delivery  $k$ ,
- ▶  $d_k$  be the number of days between delivery  $k$  and delivery  $k - 1$ ,
- ▶ and  $HDD_i$  be the heating degree day of the  $i$ th day after delivery  $k - 1$ ,  
where  $HDD = \max(0, 65 - F)$ ,  
and  $F$  is the average temperature.

Then our model is

$$D_k = \beta_0(d_k) + \beta_1 \sum_{1 \leq i \leq d_k} HDD_i \quad (1)$$

# Customers as Feature Vectors

Each customer can be represented as a vector in the feature space.  
Let...

- ▶  $lat$  be the latitude of the geographic location
- ▶  $long$  be the longitude of the geographic location
- ▶  $n$  be the number of tanks
- ▶  $c$  be the oil capacity (  $c = n * (\text{size of tanks})$  )
- ▶  $h$  denote whether oil is used to heat air or water
- ▶  $k$  be the expert estimated  $k$ -factor
- ▶  $\beta_0$  and  $\beta_1$  be the parameter estimates from eq. (1).

Then our customer can be represented as the following feature vector:

$$x = [lat, long, n, c, h, k, \beta_0, \beta_1]'$$

# Distance Metric

The results of clustering depends on a meaningful distance metric. In general our choice of distance function has the form outlined in eq. (2).

Each  $d_i$  denotes the distance functions for each feature and each  $w_i$  denotes the weight for that feature.

$$\begin{aligned} d(x, y) = & w_1 d_1(x_{lat}, x_{long}, y_{lat}, y_{long}) + w_2 d_2(x_h, y_h) \quad (2) \\ & + w_3 d_3(x_n, y_n) + w_4 d_4(x_c, y_c) \\ & + w_5 d_5(x_k, y_k) + w_6 d_6(x_\beta, y_\beta) \end{aligned}$$

Each feature measurement must be normalized so that the distances returned are comparable.

Then the weights are left free to reflect the importance of that feature.

# Feature Normalization and Distance Function

- ▶ Since  $(lat, long)$  measures real geographic distance, regular  $[0, 1]$  normalization is appropriate.
- ▶ Since  $n$  and  $h$  are binary, regular  $[0, 1]$  normalization is appropriate.
- ▶ Large outliers and non-Gaussian looking histograms in the remaining features suggest that using  $[0, 1]$  normalization or Z-score standardizations would be inappropriate.
  - ▶ These features are transformed to an *almost*  $[0, 1]$  scale.
  - ▶ The 1<sup>st</sup> percentile is mapped to 0
  - ▶ The 99<sup>th</sup> percentile is mapped to 1
- ▶ With this normalization the regular Euclidean distance metric can be used on all the features.

- ▶ Matlab's built in function, `linkage`, builds a hierarchical cluster tree using agglomerative clustering.
  - ▶ It uses the distance function described in eq. (2).
  - ▶ Each step merges clusters by minimizing the maximum distance between any pair of points from distinct clusters.
- ▶ Matlab's built in function, `cluster`, builds clusters by cutting across the dendrogram produced from `linkage` at an arbitrary 'cutoff' value.

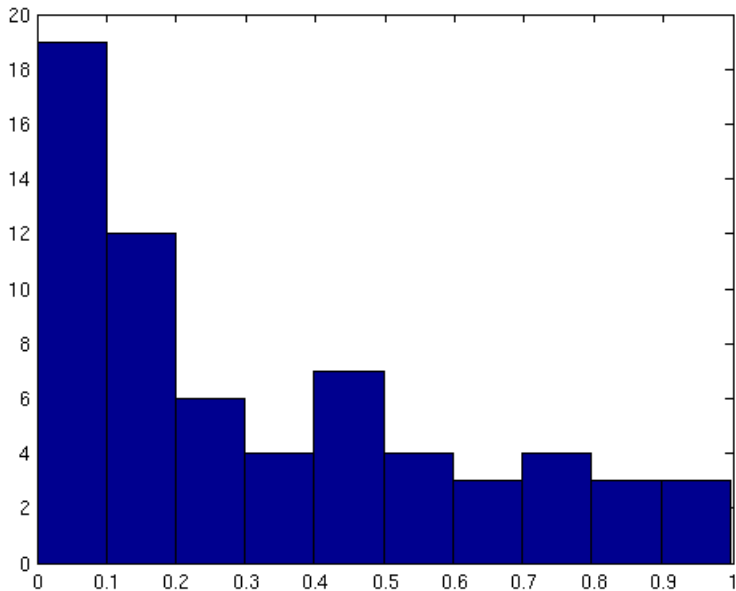
- ▶ There is no 'true' evaluation metric for an unsupervised task.
- ▶ My evaluation metric measures the effectiveness of separating customers into distinct clusters on the model from eq. (1).
  - ▶ First, the model in eq. (1) is trained on all customers collectively as a baseline.
  - ▶ Next, the model is trained on just the customers of a given cluster.
  - ▶ The sum of squared residuals is computed using both models.
  - ▶ The ratio of these values is used as an evaluation metric.

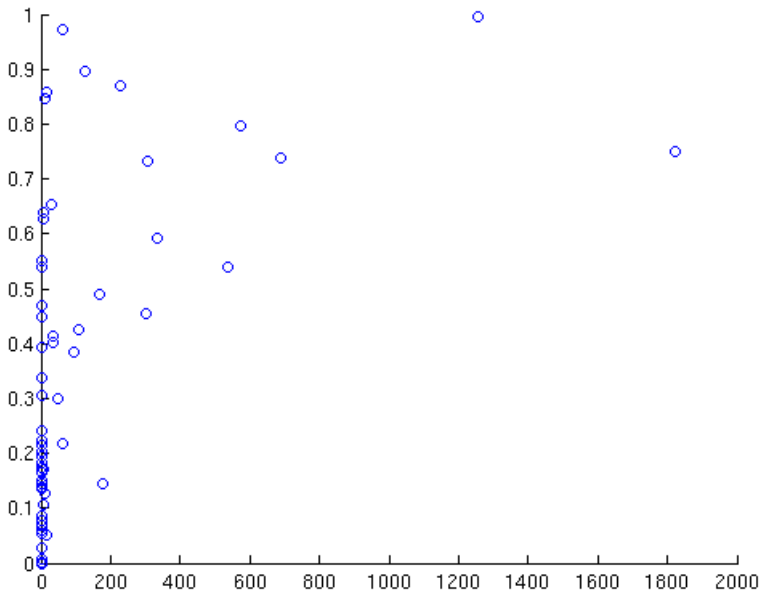
## Weights Used

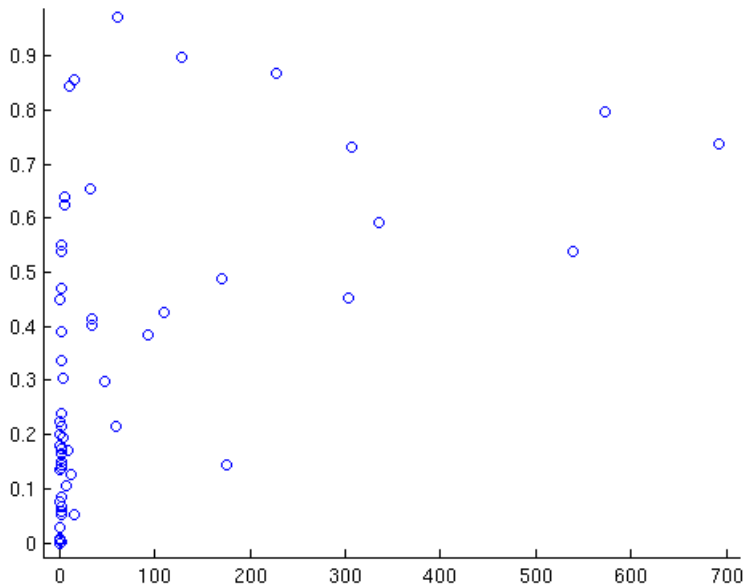
- ▶ Geographic Distance: 0.4
- ▶  $\beta_0$ : 0.2
- ▶  $\beta_1$ : 0.2
- ▶ # of Tanks: 0.03
- ▶ Oil Capacity: 0.07
- ▶ Air or Water Heating: 0.03
- ▶ k-Factor: 0.07

## Clustering Results

- ▶ Clustered with cutoff distance of 0.3
- ▶ Resulted in 65 clusters
- ▶ 30 clusters had more than 3 members
- ▶ Avg membership of clusters = 110
- ▶ Avg evaluation score of clusters = 0.32
- ▶ Min evaluation score = 0
- ▶ Max evaluation score = 1







- ▶ Weights were chosen arbitrarily.
- ▶ Cluster linking criteria chosen arbitrarily.
- ▶ Normalization scheme chosen arbitrarily.
- ▶ These decisions can be optimized under the evaluation criteria, but this task would require computational time outside the scope of my project.

# Questions?

Thanks!

Any Questions... ?