16. Claims frequency modeling with covariates extracted from telematics data

Guangyuan Gao

School of Statistics, Renmin University of China

Table of Contents

Goal and Setting

Goal: To classify the v-a heatmaps; to distinguish driving habits.
Setting:

- The v-a rectangle: $R = \bigcup_{j=1:J} R_j$.
- The set of all probability distribution on R: $\mathcal{X} \subset \mathbb{R}^J$.
- The set of driver labels: $\mathcal{N} = \{1, \ldots, n\}$.
- The v-a heatmap of driver i: $\boldsymbol{x}_i \in \mathcal{X}$.
- The set of K different categorical classes: $\mathcal{K} = \{1, \ldots, K\}$.
- A classifier function on $\mathcal{N}$: $\mathcal{C}$

$$\mathcal{C} : \mathcal{N} \to \mathcal{K}, i \mapsto \mathcal{C}(i)$$

- The dissimilarity function between $\boldsymbol{x}_b$ and $\boldsymbol{x}_c$ is defined by

$$d(\boldsymbol{x}_b, \boldsymbol{x}_c) = \frac{1}{2} \sum_{j=1}^{J} \omega_j (x_{b,j} - x_{c,j})^2, \tag{1}$$

  where ω_j are predefined weights. For simplicity we only consider $\omega \equiv 1$.
- The total dissimilarity over all drivers is defined by

$$D(\mathcal{N}) = \frac{1}{n} \sum_{b,c=1}^{n} d(\boldsymbol{x}_b, \boldsymbol{x}_c)$$

Lemma 1

The total dissimilarity over all drivers satisfies

$$D(\mathcal{N}) = \sum_{j=1}^{J} \omega_j \sum_{i=1}^{n} (x_{i,j} - \bar{x}_j)^2,$$

with empirical means $\bar{x}_j = n^{-1} \sum_{i=1}^{n} x_{i,j}$, for $j = 1, \dots, J$.

- The dissimilarity of probability masses on a sub-rectangle R_j among different drivers is measured by the empirical variance

$$s_j^2 = \frac{1}{n} \sum_{i=1}^{n} (x_{i,j} - \bar{x}_j)^2.$$

- Hence, the total dissimilarity over all drivers is given by

$$D(\mathcal{N}) = n \sum_{j=1}^{J} \omega_j s_j^2.$$

Lemma 2

The empirical means $\bar{x}_j$ are optimal in the sense that

$$\bar{x}_j = \underset{m_j}{\text{argmin}} \sum_{i=1}^{n} (x_{i,j} - m_j)^2.$$

- Introduce a regression structure by partitioning the set $\mathcal{N}$ into K clusters $\mathcal{N}_1, \ldots, \mathcal{N}_K$ satisfying

$$\cup_{k=1}^{K} \mathcal{N}_k = \mathcal{N} \text{ and } \mathcal{N}_k \cap \mathcal{N}_{k'} = \emptyset \text{ for all } k \neq k'.$$

- These K clusters define a natural classifier $\mathcal{C}$ on the set $\mathcal{N}$, given by

$$\mathcal{C} : \mathcal{N} \to \mathcal{K}, i \mapsto \mathcal{C}(i) = \sum_{k=1}^{K} k \mathbb{1}_{\{i \in \mathcal{N}_k\}}.$$

The components of total dissimilarity

$$
\begin{aligned}
D(\mathcal{N}) &= \sum_{k=1}^{K} \sum_{j=1}^{J} \omega_j \sum_{i \in \mathcal{N}_k} (x_{i,j} - \bar{x}_j)^2 \\
&= \sum_{k=1}^{K} \sum_{j=1}^{J} \omega_j \sum_{i \in \mathcal{N}_k} (x_{i,j} - \bar{x}_{j|k})^2 + \sum_{k=1}^{K} N_k \sum_{j=1}^{J} \omega_j (\bar{x}_{j|k} - \bar{x}_j)^2 \\
&= \sum_{k=1}^{K} W_k(\mathcal{C}) + B(\mathcal{C}).
\end{aligned}
$$

where $N_k = |\mathcal{N}_k|$ is the number of drivers in $\mathcal{N}_k$ and the empirical means on $\mathcal{N}_k$ are given by

$$
\bar{x}_{j|k} = \frac{1}{N_k} \sum_{i \in \mathcal{N}_k} x_{i,j}
$$

We are trying to find a classifier $\mathcal{C}$ to minimize the total within-cluster sum of squares (total within-cluster dissimilarity)

$$W(\mathcal{C}) = \sum_{k=1}^{K} W_k(\mathcal{C})$$

K-means Algorithm

1. Choose an initial classifier $\mathcal{C}^0 : \mathcal{N} \to \mathcal{K}$ with corresponding empirical means $(\bar{x}^0_{j|k})_{j,k}$.
2. Repeat for $l \geq 1$ until no changes are observed:
   1. given the present empirical means $(\bar{x}^{l-1}_{j|k})_{j,k}$ choose the classifier $\mathcal{C}^l : \mathcal{N} \to \mathcal{K}$ such that for each driver $i \in \mathcal{N}$ we have
   $$\mathcal{C}^l(i) = \underset{k \in \mathcal{K}}{\operatorname{argmin}} \sum_{j=1}^{J} \omega_j (x_{i,j} - \bar{x}^{l-1}_{j|k})^2$$
   2. calculate the empirical means $(\bar{x}^l_{j|k})_{j,k}$ on the new partition induced by classifier $\mathcal{C}^l$.

The R function

`kmeans(x, centers, nstart)` applies the K-means algorithm, where

- `x` is the $n \times J$ design matrix, containing the n drivers' heatmaps. The i, j cell is $x_{i,j}$, the probability mass on R_j of the driver i.
- `centers` is the number of cluters, i.e., K.
- `nstart` is the number of initial classifiers $\mathcal{C}^0$.

The output contains

- `cluster`: the cluster to which each driver (each row) is allocated.
- `centers`: a $K \times J$ matrix of cluster centers, i.e. $(x_{j|k})_{j,k}$.
- `totss, withinss, tot.withinss, betweenss`: total sum of squares, within-cluster sum of squares for each cluster, total within-cluster sum of squares, and total between-cluster sum of squares.

$$Y_i \overset{\text{ind.}}{\sim} \text{Poisson}(\lambda_i e_i) \qquad \text{with} \tag{2}$$
$$\lambda_i = \exp\left\{\beta_0 + s_1(\text{age driver}_i) + \beta_2 \cdot \text{age car}_i + \gamma_{\mathcal{C}(i)}\right\},$$

where

- Y_i is the number of claims from driver i
- e_i is the years-at-risk of driver i
- λ_i is the claims frequency of driver i
- s_1 is a smoothing spline to address the non-linear effects of driver's age.

$\triangleright$ It shows that $K = 2$ leads to an optimal out-of-sample prediction performance.

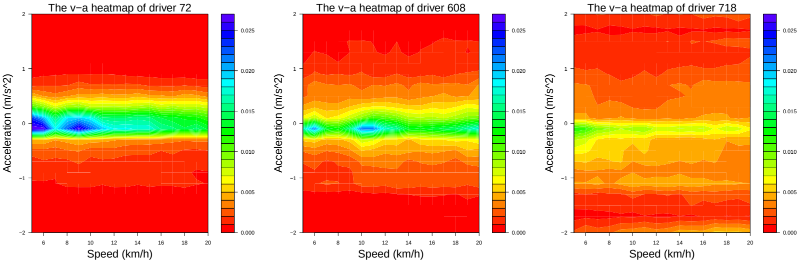

Figure 1: The v-a heatmaps $\boldsymbol{x}_i \in \mathcal{X}$ of the selected car drivers $i = 72, 608$ and 718.

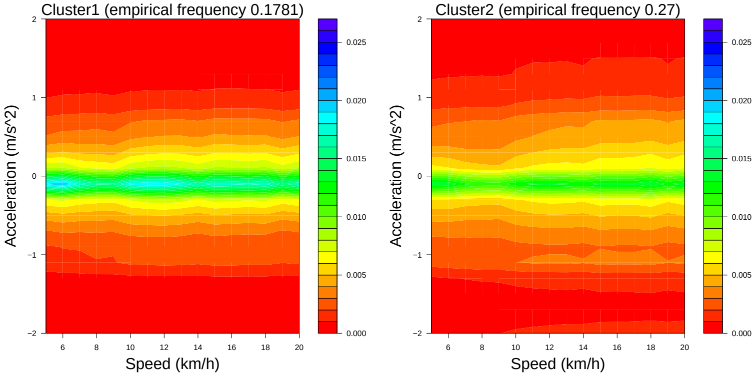

Figure 2: Average v-a heatmaps for the 2-means clusters ($K = 2$). Driver 62 and 608 are in cluster 1, and driver 718 is in cluster 2.

Denote the normalized design matrix of $\boldsymbol{X}$ by $\boldsymbol{X}^0$ (all column means are set to zero and variances are normalized to one).

Theorem 1

There exists an $n \times J$ orthogonal matrix $\boldsymbol{U}$, a $J \times J$ orthogonal matrix $\boldsymbol{V}$ and a $J \times J$ diagonal matrix $\boldsymbol{\Lambda} = \text{diag}(g_1, \ldots, g_J)$ with singular values $g_1 \geq \ldots \geq g_J \geq 0$, such that we have the following singular value decomposition (SVD)

$$\boldsymbol{X}^0 = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}'.$$

- PCA is a linear method that explores the J-dimensional covariate space for the direction of the biggest variance in $\boldsymbol{X}^0$.
- The first column of $\boldsymbol{V}$ is the direction of the biggest variance in the J-dimensional covariate space. The second column of $\boldsymbol{V}$ is the direction of the second largest variance, perpendicular to the first direction.
- The columns of $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Lambda}$ are the principal components.

The R function

`prcomp(x, center, scale.)` applies the singular value decomposition, where

- `x` is the $n \times J$ design matrix, containing the n drivers' heatmaps. The i, j cell is $x_{i,j}$, the probability mass on R_j of the driver i.
- `center` is a logical value indicating whether each column of `x` should be shifted to zero.
- `scale.` a logical value indicating whether each column of `x` should be scaled to have unit variance.

The output contains

- `rotation`: the matrix of $\boldsymbol{V}$.
- `x`: the matrix of $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Lambda}$.

$$Y_i \overset{\text{ind.}}{\sim} \text{Poisson}(\lambda_i e_i) \qquad \text{with}$$
$$\lambda_i = \exp\left\{\beta_0 + s_1(\text{age driver}_i) + \beta_2 \cdot \text{age car}_i + \beta_3 \cdot P_{i,1}\right\},$$

- It turns out that only the 1st PC has a strong relationship with claims frequency.
- The effect of the 1st PC on claims frequency is log-linear.
- It turns out that the predictive power of the 1st PC is better than the traditional risk factors such as driver's age.

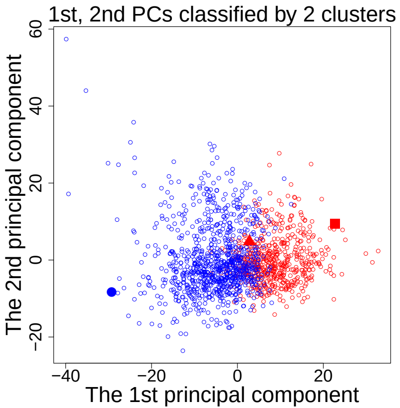

Figure 3: Red indicates cluster 1 and blue indicates cluster 2 of the 2-means clustering. Square, triangle and circle symbols indicate drivers 72, 608 and 718, respectively. Each point corresponds to one of the $n = 1,478$ drivers' heatmaps.

- It turns out that once the first principal component is considered in the claims frequency model, the clusters are no longer needed.
- This is because the first principal component is highly related to the selected clusters.
- The separation between the two clusters is almost a vertical line. So the first principal component is enough to explain the clustering of the 2-means algorithm.
- In cluster analysis we cannot distinguish driver 72 and 608, but in PCA their 1st PCs are obviously different.
- Driver 608 tends to have a higher claims frequency than driver 72.

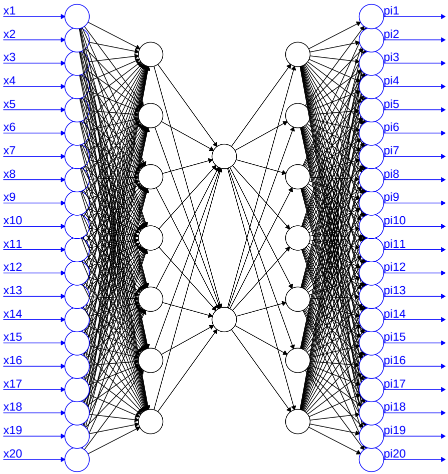

Figure 4: Deep neural network with (p, q, p) hidden neurons.

Autoencoder is a bottleneck neural network. It is also called as non-linear PCA since non-linear activation functions such as sigmoid or hyperbolic are involved.

- An autoencoder consists of an encoder $\varphi : \mathcal{X} \to \mathcal{Z}$, where $\mathcal{Z}$ is low-dimensional, and of a decoder $\psi : \mathcal{Z} \to \mathcal{X}$.
- The goal of is to choose these functions φ and ψ such that the output $\pi(\boldsymbol{x}) = \psi \circ \varphi(\boldsymbol{x})$ is close to the input $\boldsymbol{x}$.
- The value $\varphi(\boldsymbol{x}) \in \mathcal{Z}$ is then used as a low-dimensional representation for $\boldsymbol{x} \in \mathcal{X}$.
- The encoder will map $\boldsymbol{x} \in \mathcal{X}$ to $\boldsymbol{z}^{(2)} = \boldsymbol{z}^{(2)}(\boldsymbol{x}) \in \mathcal{Z} = [-1, 1]^2$.
- The symmetry of (p, q, p) has major advantages in calibration of the corresponding encoding functions.

- The first hidden layer is given by

$$z_l^{(1)}(\boldsymbol{x}) = \tanh\left(w_{l,0}^{(1)} + \sum_{j=1}^{J} w_{l,j}^{(1)} x_j \right), \qquad \text{for } l = 1, \ldots, p, \tag{3}$$

- The second hidden layer by

$$z_l^{(2)}(\boldsymbol{x}) = \tanh\left(w_{l,0}^{(2)} + \sum_{j=1}^{p} w_{l,j}^{(2)} z_j^{(1)}(\boldsymbol{x}) \right), \qquad \text{for } l = 1, \ldots, q. \tag{4}$$

- This provides the encoder $\varphi(\boldsymbol{x}) = \boldsymbol{z}^{(2)}(\boldsymbol{x}) = (z_1^{(2)}(\boldsymbol{x}), z_2^{(2)}(\boldsymbol{x}))' \in [-1,1]^2$ for bottleneck $q = 2$.

- The third hidden layer of the neural network is given by

$$z_l^{(3)}(\boldsymbol{x}) = \tanh\left(w_{l,0}^{(3)} + \sum_{j=1}^{q} w_{l,j}^{(3)} z_j^{(2)}(\boldsymbol{x})\right), \qquad \text{for } l = 1, \ldots, p, \tag{5}$$

- This is then used in the regression equations

$$\mu_j(\boldsymbol{x}) = \mu_j(\boldsymbol{x}; \boldsymbol{\alpha}^{(j)}) = \alpha_0^{(j)} + \sum_{l=1}^{p} \alpha_l^{(j)} z_l^{(3)}(\boldsymbol{x}), \qquad \text{for } j = 1, \ldots, J. \tag{6}$$

- Functions (5)-(6) provide the decoder defined by the following multinomial logistic probabilities $\pi(\cdot) = (\pi_j(\cdot))_{j=1:J}$ with

$$\pi_j(\boldsymbol{x}) = \frac{\exp\{\mu_j(\boldsymbol{x})\}}{\sum_{j'=1}^{J} \exp\{\mu_{j'}(\boldsymbol{x})\}}, \qquad \text{for all } \boldsymbol{x} \in \mathcal{X}. \tag{7}$$

The R function

- We apply the gradient decent method to calibrate the weights $w^{(1)}, w^{(2)}, w^{(3)}, \alpha$.
- One needs to make significant efforts to train a neural network.
- R interface to `keras` might be helpful.

$$Y_i \overset{\text{ind.}}{\sim} \text{Poisson}(\lambda_i e_i) \qquad \text{with}$$
$$\lambda_i = \exp\left\{\beta_0 + s_1(\text{age driver}_i) + \beta_2 \cdot \text{age car}_i + \beta_3 \cdot z_0(\boldsymbol{x}_i)\right\},$$

with the transformed bottleneck neuron

$$z_0(\boldsymbol{x}_i) = z_1^{(2)}(\boldsymbol{x}_i) - 0.5 z_2^{(2)}(\boldsymbol{x}_i), \tag{8}$$

- Investigation indicates that both bottleneck neurons are simultaneously needed in the model
- The effects of both bottleneck neurons are log-linear.
- The estimated coefficients are 3.52 and -1.78, which motives (8).

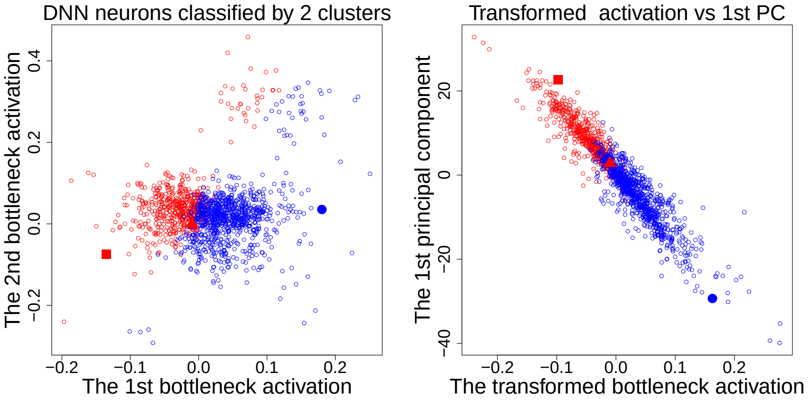

Figure 5: Red indicates cluster 1 and blue indicates cluster 2 of the 2-means clustering. Square, triangle and circle symbols indicate drivers 72, 608 and 718, respectively. Each point corresponds to one of the $n = 1,478$ drivers' heatmaps.

- The separation line in the first plot indicates that both bottleneck neurons are related to 2-means clusters.
- The second plot shows a strong linear relationship between the 1st PCs and the transformed bottleneck neurons.
- We do not need the 1st PCs and the transformed bottleneck neurons in the model simultaneously.

Exercises:

- Besides equation (1), list another two dissimilarity functions which might be used in the clustering analysis.
- Discuss how to recover the heatmap for each driver using the first principal components.
- Explain why we need equation (7).