Churn Modelling: A Perspective on Mixture-Based Clustering of Customer Behaviours

Victor (Sheng) Wang

University of Oxford

Supervisors: Andrew Mellor, Junaid Mubeen

A technical report for

InFoMM CDT Mini-Project 1

in partnership with

Whizz Education

Trinity 2018

Contents

1 Introduction

Retained customers in general create higher revenues than new customers do, and making a sell to a new customer can cost up to 5 times more depending on the business [5]. Therefore, many companies form the Customer Relationship Management (CRM) team with a focus on customer retention strategies. A crucial step is then to identify high risk customers who are intending to discontinue their usage of the services. This assessment is better known as *churn prediction*.

Our project aims to perform the churn prediction task based on investigation of customers's behavioural clusterings, and formulate the processes into a scalable pipeline which can be easily reused, updated and extended for many applications. In particular, we apply the pipeline to analyse pupil subscribers' data for Whizz Education (referred to as "Whizz"). Whizz provides online virtual tutorial service, Math-Whizz, which pupils can access by purchasing subscriptions. We have investigated whether behavioural-based data analytics can be used to help detect potential subscription cancellations.

The pipeline starts from representing pupils' behaviours by numerical data, also known as *feature* extraction. The structured features' data are fed into a mixture model that decodes features' distribution as a weighted combination of simple distributions. Since simple distribution is assumed to be generated by an unobserved *state*, we are interested in uncovering the states as well as their associated emitted feature distributions, also known as *clusters*. The mixture model is deemed to be effective if the identified clusters of pupils exhibit distinguishable proportions of churn, or *churn rates*. Assessing churn rate of the fitted mixture model results in a trained model that establishes a map between features to probability of churn. This enables the prediction of new coming pupils' churn probabilities by feeding their feature data into the trained model.

We show that clusters inferred from behavioural data are characterised by non-trivially different churn rate. Moreover, there is recognisable pattern observed in cluster sizes. This verifies the effectiveness of the mixture model approach in uncovering what level of churn risk the pupils are at. We further investigate how features impact churn probability and examine that the mixture model can result in very minimal overfitting issues in prediction practices. We also infer the temporal transitional probabilities of states by studying on the dynamics of behavioural changes.

The report is structured as follows. We start by elaborating the modelling framework and pipeline in section 2. Next in section 3 we describe the feature data preparation and pre-processing techniques used to adapt Whizz's data for better modelling performance. Then we explain the details of clustering analysis and model evaluation in section 4. Finally, we carry out prediction task and also assess overfitting issues, as detailed in section 5. We conclude the project with deliverables and future directions in section 6.

2 Modelling Customer Behaviours

Customers' behaviour evolves over time as they respond to business offering and adjust to their own demand. We would like to represent the various behaviour dynamics of a collection of customers by panel data, and employ the Markov chain model to describe these data. In addition, we use the mixture model to find Markov states, each of which defines a partition of the behaviours observable within a single time interval.

2.1 Representing Behaviours by Features

A *feature* is an individual measurable property of a behaviour being observed, and choosing informative features is crucial for effective clustering. For example, to measure how often the pupil uses Whizz online tutorial, we can define the time spent or number of visits within a month as the feature. For each customer, we define multiple features to capture his behaviours of various aspects within a time interval. Suppose we are interested in studying behaviours of n customers in T discrete consecutive time periods, and define m features, then we denote the feature data as a sequence:

$$\{\mathbf{X}_1,\ \mathbf{X}_2,\ \ldots,\ \mathbf{X}_t,\ \ldots,\ \mathbf{X}_T\}\,, \tag{1}$$

in which the t-th element $\mathbf{X}_t = (x_{ij}^t) \in \mathbb{R}^{m\times n}$ is a matrix for all $t = 1, 2, \ldots, T$. In addition, we denote the features of customer j at the t-th time interval by the j-th column of $\mathbf{X}_t$ by $\mathbf{x}_{tj} = [x_{1j}^t\ x_{2j}^t\ \cdots\ x_{mj}^t]^\top$.

2.2 Customer Journeys and Markov Chain

Customer journeys reflect their feature dynamics over time, denoted as (1). One practical challenge of computing such sequence of matrices is to resolve the inconsistency present in journeys of different customers. The inconsistency refers to the problem that the time intervals for different customers being alive in the services are not aligned, so that their features are not comparable. Moreover, it is highly likely to have significant missing information within specific time intervals for customers who have not yet entered the service or have already churned. To resolve the inconsistency, we align and aggregate customers' features by *customer month* rather than calendar month. Doing so enables the effective modelling using Markov chain.

2.2.1 Customer Month

Splitting pupils' behaviours into monthly time periods makes most sense provided the business settings at Whizz. Pupils subscribe to access Whizz products on a 1-month contract, and make the choice to leave the service at the end of each subscription. If no action is taken, a renewal will be made by default.

Due to the inconsistency present in journeys of different customers, we align their features by switching the reference from calendar month to customer month. This is illustrated by an example in Figure 2.1. It provides much cleaner and more sensible data for modelling task.

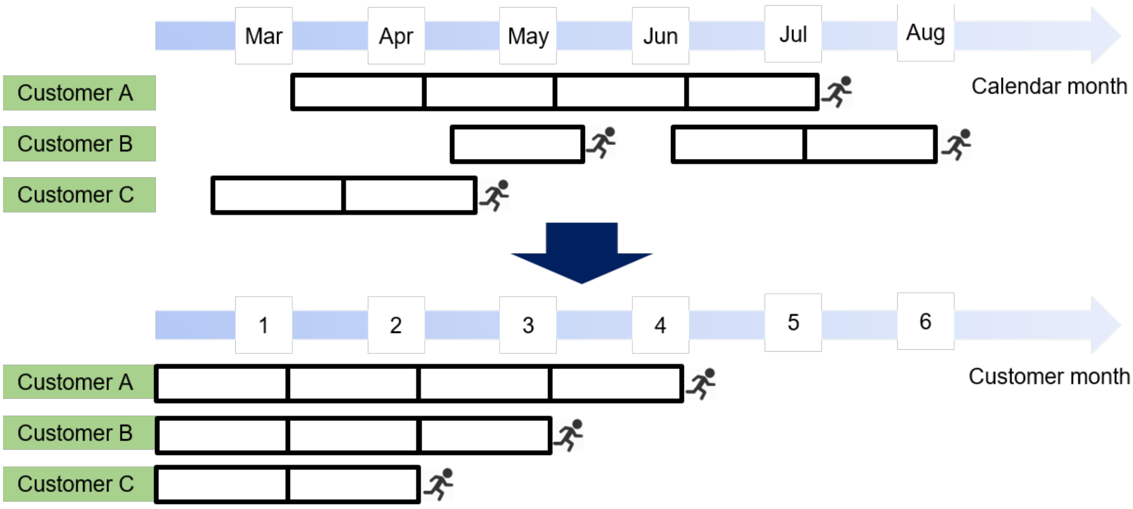

Figure 2.1: Change reference from calendar month to customer month. Customer A, B and C have very different journeys in the sense of subscription start and end dates. Each block represents customer's monthly features. Under calendar month reference, we have to choose studying months from March to August to cover all activities. This choice results in irregular temporal distribution of missing information for all 3 customers. After changing the reference to customer month, features are aligned by customer month and therefore comparable. Moreover, the missing information only occurs after the customer churns. It can also handle discontinuous subscriptions like the case of customer B.

2.2.2 Markov Chain - Dynamic Model for Behavioural Changes

We assume that customers with intentions to churn exhibit different behaviours than others do. Behaviours are different distributionally, and generated from a finite number of *states*. Then the discrete time behaviour of each customer results in a chain of states over time. This formulates into a discrete-time Markov chain with states transiting over time, where each state emits distinguishable behaviour distribution from others. An example is given in Figure 2.2. In brief, a Markov chain is a stochastic process characterised by *transition* and *emission* probabilities.

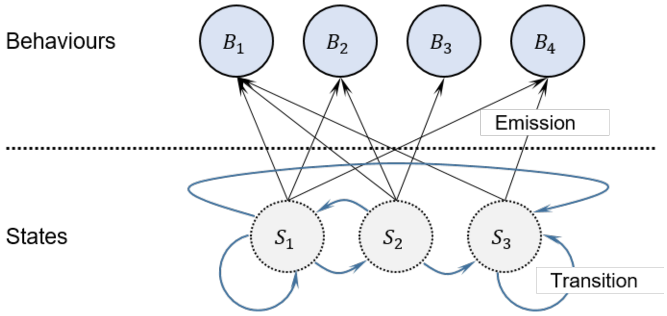

Figure 2.2: Behaviours emitted from Markov states. States S_1, S_2 and S_3 generate differently distributed behaviours. For example, S_1 generates $\{B_1, B_2, B_4\}$ while S_3 produces $\{B_1, B_4\}$. Even if the two states can generate the same set of behaviours, the emission probabilities can be different, thus still resulting in different behaviour distributions. States transit between each other over time stochastically.

Transition Consider the behavioural journey of customer j, which is represented by a sequence of feature data $\{\mathbf{x}_{tj}\}_{t=1}^{T}$. At time interval t, feature $\mathbf{x}_{tj}$ is an instance from a distribution generated by a state. Let's denote the time sequence of generative states as $\{s_t(j)\}_{t=1}^{T}$. In a Markov chain setting the sequences are defined stochastically, with the next state being conditionally dependent on the present state, but not any further previous history. This is known as *Markov property*.

If we define a finite set of states by $\mathcal{S} = \{S_q\}_{q=1}^{Q}$, then $s_t(\cdot)$ is a map from $\mathbb{N}$ to $\mathcal{S}$. Due to the Markov property, successive states are linked together with the (conditional) *transition probability matrix* defined as:

$$A_t = (a_{pq}^{t}) \in [0,1]^{Q\times Q}, \tag{2a}$$

with

$$a_{pq}^{t} = \mathbb{P}\left(s_{t+1} = S_p | s_t = S_q\right). \tag{2b}$$

If we assume time-homogeneity or *stantionarity* of the Markov chain, then we can remove the time subscript from the notations because the parameters describing all probabilistic transitions are themselves constant.

Emission Recall that feature data $\{\mathbf{x}_{tj}\}_{t=1}^{T}$ observed are instances from some distribution. We introduce a generic notation for observed sequence of features, $\{\mathbf{o}_t\}_{t=1}^{T}$, where $\mathbf{o}_t = [o_{t1}\ o_{t2}\ \cdots\ o_{tm}]^{\top}$. The sequence of observations are generated in the following way:

1. At each time step, the system generates a state s_t according to the state-to-state transition probability matrix A_t.

2. Once the state s_t has been generated, the system generates a cluster c_t according to a state-to-cluster emission probability distribution $\pi(s_t, c_t)$. Suppose we define a finite collection of clusters $\mathcal{C} = \{C_k\}_{k=1}^{K}$, then we denote:

$$\pi_{qk} = \pi(S_q, C_k) = \mathbb{P}\left(c_t = C_k | s_t = S_q\right). \tag{3}$$

3. Once the cluster c_t have been determined, an observation vector $\mathbf{o}_t$ is produced probabilistically according to some cluster-specific distribution $\phi(\mathbf{o}_t|\theta(s_t, c_t))$, where $\theta(s_t, c_t)$ denotes the distribution parameters. We write,

$$\phi(\mathbf{o}_t|\theta_{qk}) = \phi(\mathbf{o}_t|\theta(S_q, C_k)) = \mathbb{P}\left(\mathbf{o}_t | s_t = S_q, c_t = C_k\right). \tag{4}$$

Given the generative process described above, we can now model the state-to-observation emission probability by a mixture of densities:

$$\begin{aligned} b_q(\mathbf{o}_t) &= \mathbb{P}\left(\mathbf{o}_t | s_t = S_q\right) && \text{(5a)}\\ &= \sum_{k=1}^{K} \mathbb{P}\left(c_t = C_k | s_t = S_q\right) \mathbb{P}\left(\mathbf{o}_t | s_t = S_q, c_t = C_k\right) && \text{(5b)}\\ &= \sum_{k=1}^{K} \pi_{qk}\phi(\mathbf{o}_t|\theta_{qk}). && \text{(5c)} \end{aligned}$$

2.2.3 Decoding the Markov Chain

The problem is how to estimate the transition probabilities and parameters in the emission term, A_t and (π, θ) from the observations $\mathbf{X}_t$. Our strategy is to split this decoding process into two separate steps, where we first make inference on emission and then uncover the temporal structure configured by transition.

Decoding Emission Ideally from the perspective of practical application of the model, we wish to encode a finite number of states representing different levels of churn risk. As a consequence, the prediction task is to find out the state that the pupil belongs to and therefore assigning the associated risk label. Hence, we choose not to infer states purely from observed behavioural data, but define states by also incorporating churn outcome information. At high level, we take two steps to estimate the parameters in the emission term:

1. We look at the behavioural distribution without conditioning on state, namely,

$$b(\mathbf{o}_t) = \mathbb{P}(\mathbf{o}_t) = \sum_{k=1}^{K} \mathbb{P}(c_t = C_k)\mathbb{P}(\mathbf{o}_t | c_t = C_k) = \sum_{k=1}^{K} \pi_k \phi(\mathbf{o}_t | \theta_k). \tag{6}$$

   Then we estimate $\{\pi_k, \theta_k\}_{k=1}^{K}$ from observed feature data $\mathbf{X}_t$ with pre-defined multivariate kernel density $\phi(\cdot)$. This will be elaborated in section 2.3.

2. The previous step gives not only the weights π and cluster density $\phi(\cdot|\theta)$, but also a consequential cluster assignment of all customers. If we know the churn outcome for all customers, then we can calculate the proportion of churned customers, or churn rate, within each cluster. Thereafter, we can form states by grouping together clusters of similar level of churn rate.

   Formally, we define the set of customers who have been assigned into cluster C_k as

$$\mathcal{N}_k^t = \{j : \ j \text{ is assigned into } C_k \text{ at time } t\}. \tag{7}$$

   Meanwhile, we define the churn outcome information by a set

$$\mathcal{N}_{\text{churn}}^t = \{j : \ j \text{ churns at } t+1\}, \tag{8}$$

   so that the cluster churn rate is defined as

$$\lambda_k^t = \frac{|\mathcal{N}_k^t \cap \mathcal{N}_{\text{cancel}}^t|}{|\mathcal{N}_k^t|}. \tag{9}$$

   Once the churn rates of all K clusters, $\{\lambda_k^t\}_{k=1}^{K}$, are computed, we group them and form Q states. Typically Q is much smaller than K. We denote the set of clusters consisting state S_q as,

$$\mathcal{K}_q^t = \{k : \ C_k \text{ is emitted from } S_q \text{ at time } t\}. \tag{10}$$

Afterwards, we can revisit the calculation for the state-to-observation emission probability $b_q(\mathbf{o}_t)$. Note that by the way we define state, we have

$$\pi_{qk} = \mathbb{P}\left(c_t = C_k | s_t = S_q\right) = \frac{|\mathcal{N}_k^t|}{\sum_{l \in \mathcal{K}_q^t} |\mathcal{N}_l^t|} \mathbb{1}_{\{k \in \mathcal{K}_q^t\}}, \tag{11}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function. Then

$$b_q(\mathbf{o}_t) = \sum_{k=1}^{K} \pi_{qk} \phi(\mathbf{o}_t; \theta_{qk}) = \sum_{k \in \mathcal{K}_q^t} \frac{|\mathcal{N}_k^t|}{\sum_{l \in \mathcal{K}_q^t} |\mathcal{N}_l^t|} \phi(\mathbf{o}_t | \theta_k). \tag{12}$$

Decoding Transition We define the set of customers who transit from S_p at t to S_q at $t+1$ as

$$\mathcal{Q}_{q \to p}^t = \left\{ j : \ s_t(j) = S_q, \ s_{t+1}(j) = S_p \right\}. \tag{13}$$

We assume that customers' behaviours are i.i.d. samples from the generating process, then the maximum likelihood estimate of the transition probability is

$$\hat{a}_{pq}^t = \frac{|\mathcal{Q}_{q \to p}^t|}{\sum_{l=1}^{Q} |\mathcal{Q}_{q \to l}^t|}. \tag{14}$$

If the Markov chain is assumed to be stationary, then we estimate

$$\hat{a}_{pq} = \frac{\sum_{t=1}^{T-1} |\mathcal{Q}_{q \to p}^t|}{\sum_{t=1}^{T-1} \sum_{l=1}^{Q} |\mathcal{Q}_{q \to l}^t|}. \tag{15}$$

2.3 Probabilistic Clustering Using Mixture Model

A critical step of the state-cluster-observation Markov chain is the mixture model that describes the probabilistic assignment of observations to clusters. Practical calibration of the mixture model faces many choices such as the kernel density $\phi(\cdot)$, the number of clusters K, etc. A fundamental model setting is however the choice of frequentist or Bayes approach to estimate model parameters. We choose specifically the *Dirichlet Process Mixture* setting which has two important features:

1. It is Baysian and treats θ as a random variable, of which distributions will be updated from a prior as observed data coming in.

2. The Dirichlet process (DP) is used as a nonparametric prior resulting in that the number of clusters is random and grows as new data are observed.

The benefits of this model choice are massive. It does not view the observed data as infinitely available as independent replicates like frequentists, so that it does not worry about the unobserved data and can be updated with new data coming in. Moreover, it infers the number of clusters from observed data, and opens the opportunities of finding new clusters as more data are observed.

2.3.1 Dirichelet Process Mixture Model

Definitions A Dirichelet Process (DP) is a distribution of a random measure. Let G_0 be a base distribution (measure) for our cluster density parameter $\theta \in \Theta$, a measurable space, and let α be a positive, real-valued scalar. A random measure G is then distributed according to *Dirichelet Process* with scaling parameter α and base measure G_0:

$$G \sim \mathrm{DP}(\cdot|G_0, \alpha), \tag{16a}$$

if for all $K \in \mathbb{N}$, and all $\{\Theta_1, \dots, \Theta_K\}$ finite partitions of Θ:

$$(G(\theta_1), \dots, G(\theta_K)) \sim \mathrm{Dir}\left(\alpha G_0(\theta_1), \dots, \alpha G_0(\theta_K)\right), \tag{16b}$$

where $\mathrm{Dir}(\cdot)$ denotes the *Dirichelet distribution.* The Dirichelet distribution is a distribution of the standard $K-1$ simplex. Let $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^K$ with $\sum_{k=1}^K \pi_k = 1$ and $\forall k : \pi_k \geq 0$, and let $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$ with $\alpha_1, \dots, \alpha_K \geq 0$. Then

$$\mathbb{P}(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \mathrm{Dir}(\alpha_1, \dots, \alpha_K) = \frac{1}{\mathrm{Beta}(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} = \frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}, \tag{16c}$$

where $\mathrm{Beta}(\cdot)$ is the beta function, $\Gamma(\cdot)$ is the gamma function.

Clustering Effect We use DP as a prior to distribution of cluster parameter θ:

$$\theta|G \sim G(\cdot) \quad \text{and} \quad G \sim \mathrm{DP}(\cdot|G_0, \alpha). \tag{17}$$

This model exhibits a "clustering effect" which enables us to infer number of clusters from data rather than pre-defining it. Suppose we independently draw n random values $\theta^{(j)}$ from G under the model (17), then Blackwell and MacQueen's urn representation theorem [1] states that, marginalising out the random measure G, the joint distribution of the collection of variables $\{\theta^{(1)}, \dots, \theta^{(n)}\}$ exhibits a clustering effect:

$$\mathbb{P}\left(\theta^{(j)}|\theta^{(1)}, \dots, \theta^{(j-1)}\right) \propto \alpha G_0(\theta^{(j)}) + \sum_{l=1}^{j-1} \delta_{\theta^{(l)}}(\theta^{(j)}), \tag{18}$$

where $\delta_{\theta^{(l)}}(\cdot)$ is a Dirac delta at $\theta^{(l)}$. Thus the variables $\{\theta^{(1)}, \dots, \theta^{(n)}\}$ are randomly partitioned according to which variables are equal to the same value. Moreover, let $\{\theta_1, \dots, \theta_K\}$ denote the distinct values of the drawn samples $\{\theta^{(1)}, \dots, \theta^{(j-1)}\}$, let $\{\kappa_1, \dots, \kappa_{j-1}\}$ be the assignment variables such that $\theta^{(l)} = \theta_{\kappa_l}$. Then,

$$\mathbb{P}\left(\theta^{(j)}|\theta^{(1)}, \dots, \theta^{(j-1)}\right) \propto \frac{\alpha}{j-1+\alpha} G_0(\theta^{(j)}) + \sum_{k=1}^{K} \frac{|\{l : \kappa_l = k\}|}{j-1+\alpha} \delta_{\theta^{(l)}}(\theta^{(j)}). \tag{19}$$

This implies that the j-th draw has a probability of $(j-1)/(j-1+\alpha)$ to be exactly the same as some previously drawn value. This forms a natural clustering effect. In addition, with a probability of $\alpha/(j-1+\alpha)$ a new cluster will be produced with the new distinct value θ_{K+1}. Hence the number of clusters is allowed to grow as new data are observed.

Dirichelet Process Mixture Model Given the clustering effect exhibited in DP, we define the *Dirichelet process mixture model* as:

$$\mathbf{o}|\theta \sim \phi(\cdot|\theta) \quad \text{and} \quad \theta|G \sim G(\cdot) \quad \text{and} \quad G \sim \mathrm{DP}(\cdot|G_0, \alpha), \tag{20}$$

recall that $\mathbf{o}$ is the observed feature vector, we remove the time subscript t since the mixture model holds for all time steps.

2.3.2 Generative Process with Stick-Breaking Representation

The definition of DP stated in (16) does not provide useful information of generating a DP in practice. Sethuraman [4] proposes the *stick-breaking representation* to explicitly construct a DP. Suppose there are two infinite sets of independent random variables, $\beta_k \sim \mathrm{Beta}(1, \alpha)$ and $\theta_k \sim G_0$, $\forall k \in \{1, 2, \dots\}$. The stick-breaking representation of G is:

$$G(\cdot) = \sum_{k=1}^{\infty} \pi_k(\boldsymbol{\beta}) \delta_{\theta_k}(\cdot), \tag{21a}$$

where

$$\pi_k(\boldsymbol{\beta}) = \beta_k \prod_{l=1}^{k-1} (1 - \beta_l). \tag{21b}$$

Note that by construction $\sum_{k=1}^{\infty} \pi_k(\boldsymbol{\beta}) = 1$. In the DP mixture model, $\boldsymbol{\pi}(\boldsymbol{\beta}) = \{\pi_k(\boldsymbol{\beta})\}_{k=1}^{\infty}$ gives the mixing proportions of mixture components represented by atoms $\{\theta_k\}_{k=1}^{\infty}$. By far, we can describe the feature data generative process as follows:

1. Draw an infinite collection of $\beta_k|\alpha \sim \mathrm{Beta}(1, \alpha)$, $\forall k \in \{1, 2, \dots\}$.
2. Draw an infinite collection of $\theta_k|G_0 \sim G_0$, $\forall k \in \{1, 2, \dots\}$.
3. For j-th data point, $j = 1, \dots, n$:
   (a) Draw $z^{(j)}|\boldsymbol{\beta} \sim \mathrm{Mult}(\boldsymbol{\pi}(\boldsymbol{\beta}))$;
   (b) Draw $\mathbf{o}^{(j)}|z^{(j)} \sim \phi(\mathbf{o}|\theta_{z^{(j)}})$.

2.3.3 Inference

We use the variational inference for the Dirichelet process mixture model presented by Blei and Jordan [2]. In practice Dirichlet Process inference algorithm is approximated and uses a truncated distribution with a fixed maximum number of components, say $K_{\max}$, of β and θ. Nevertheless, the number of components actually used K almost always depends on the data, and that $K \leq K_{\max}$.

2.3.4 Predictive Density

Based on model setting (20), model configuration $(\phi(\cdot), \alpha, G_0)$ and observed feature data points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, we are able to make inference on the posterior distribution $\mathbb{P}(\theta|\mathbf{x}_1, \dots, \mathbf{x}_n, \alpha, G_0)$. Afterwards, we can compute the predictive density:

$$\mathbb{P}(\mathbf{x}|\mathbf{x}_1, \dots, \mathbf{x}_n, \alpha, G_0) = \int \phi(\mathbf{x}|\theta) \mathbb{P}(\theta|\mathbf{x}_1, \dots, \mathbf{x}_n, \alpha, G_0) \mathrm{d}\theta. \tag{22}$$

2.4 Modelling Pipeline

We summrise the processes of our modelling framework as a pipeline displayed in Table 1. We will apply this pipeline to build a churn prediction model for Whizz.

No.	Process	Input	Output
1	**Feature extraction**: extract informative feature data from historical records of pupils' activity, and represent them in the suitable data structure that can be fed into the behavioral model.	Raw data from Whizz database that records pupils' ID, subscription history, activity history, etc.	Feature data represented by $\{\mathbf{X}_t\}_{t=1}^T$, as defined in (1).
2	**Feature distributional modelling**: choose the most suitable distributions for the all m features to fit. Independent features can be modelled separately, while correlated features shall be modelled as a part of a multivariate distribution.	Feature data represented by $\{\mathbf{X}_t\}_{t=1}^T$.	Density form $\phi(\cdot)$, as defined in (3). (Note in this step we only propose the density form but not fit the distribution parameters.)
3	**Fitting DP mixture model**: by assuming behaviours are generated from a Dirichelet process mixtures, make inference on parameters based on observed features.	Feature data represented by $\{\mathbf{X}_t\}_{t=1}^T$, feature density form $\phi(\cdot)$, and DP mixture model defined in (20).	Posterior distribution $\mathbb{P}(\theta\|\mathbf{X}, \alpha, G_0)$, a collection of clusters parametrised by different θ.
4	**Fitting Markov chain**: use the identified clusters as well as churn outcome information to define states; uncover their transition probabilities.	Identified probabilistic clusters with density $\{\phi(\cdot\|\theta_k)\}_{k=1}^K$; churn outcome.	A finite set of states $\mathcal{S} = \{S_q\}_{q=1}^Q$ and transition probability matrix $\{A_t\}_{t=1}^T$
5.1	**Analytics on behaviours**: study the properties of pupils' behaviours such as the temporal transition probabilities, how each feature impact the level of churn risk, etc.	States $\{S_q\}_{q=1}^Q$, transition $\{A_t\}_{t=1}^T$, cluster churn rate $\{\lambda_k^t\}_{t,k}$ defined in (9), etc.	Temporal state transition analysis, feature analysis, etc.
5.2	**Prediction on new pupils**: predict the level of churn risk for new pupils based on their behaviours with the observation-cluster-state probabilistic assignment trained from previous steps.	States $\{S_q\}_{q=1}^Q$, clusters parametrised by $\mathbb{P}(\theta\|\mathbf{X}, \alpha, G_0)$ for $\theta \in \{\theta_k\}_{k=1}^K$, cluster-state assignment.	State assignment for new pupils as well as their associated level of churn risk.

Table 1: Modelling pipeline showing processes in sequence, along with the input and output of each modular process. The last two process 5.1 and 5.2 are independent from each other and are performed for different purposes.

3 Data Description and Pre-processing

We will explain how we extract features from pupils' activity data (first process in the modelling pipeline in Table 1). Before that, we describe the data used in our model by summary statistics. The extracted feature data will then be pre-processed to make their distributions more suitable for the statistical inference on mixture model.

3.1 Scope of Data

Whizz stores and maintains data generated from business activities in a database consisting of several relational tables. The most relevant tables we use are listed in Table 2. The study period is from January 1st, 2014 to April 20th, 2018. The number of records within this study period in each data table is also indicated.

Data Table	Description	Number of Records
Account Information	Pupils' ID and personal information such as date-of-birth.	5,685
Subscription history	Start date and end date of each new subscription or renewed associated with each pupil.	28,025
Lesson history	Details of each visit activity for each pupil during his subscription period. The visit activity includes the date of visit, time spent, score achieved, lesson outcome, etc.	1,640,080

Table 2: Description of data tables. The number of records within the study period 2014-01-01 ∼ 2018-04-20 in each data table is also indicated.

We need refine the data set to accommodate the model needs. These include:

1. The cluster-to-state assignment requires churn outcome information to calculate churn rate for each cluster, as shown in (9). Therefore, we remove data for 1234 pupils who still have active subscription as of the end of our study period.

2. To define customer month, we need to know the start date of the first subscription for pupils. Therefore, we remove data for 154 pupils whose first subscription starts before the start of our study period.

3. We restrict our study to monthly subscribers only, then we remove data for 1625 annual subscribers[1].

As a result, we finally have 2,672 pupils' data including 17,861 subscription records and 450,548 lesson history records. Therefore we will fit our model using data for $n = 2672$ customers. In addition, the subscription history records infer $T = 49$ customer months in the study period. This means that pupils with longest subscription cancel after 49 months of usage.

3.2 Feature Extraction

Based on the scope of data, we define features from lesson history records. Since we split pupil's activities into monthly time period, then all features measure monthly-aggregated behaviours. We describe the feature name, measurement and value range in Table 3.

[1] There is no strict separation between monthly subscribers and annual subscribers, because in practice pupils can switch between monthly and annual subscription types. Our study keeps pupils who have only committed to monthly subscriptions during the whole lifetime.

Indeed, Whizz's data records contain rich information and the features listed in Table 3 are only a part of them. We view these features as most straightforward and relevant by consulting industrial expertise from staff at Whizz.

Feature	**Description**	**Value**
Number of visits	Number of visits to the online tutorial system within a customer month.	Non-negative integer
Usage time	Time spent in the online tutorial system within a customer month.	Non-negative integer, measured in seconds
Time since last visit	Number of days since the last visit to the online tutorial system.	Non-negative integer, measured in days
Number of helps	Number of helps the pupils have asked for during tutorials within the customer month.	Non-negative integer
Progress level	Overall progress the pupils have achieved at the end of the customer month. Whizz measures progress of pupils by counting the number of exercises/tests passed since their first subscription. The initial progress is 0.	Non-negative integer
Progress delta	Progress increment the pupils have achieved during the customer month.	Non-negative integer
Effective progress	Average progress increment per hour the pupils will make during the customer month.	Non-negative real number
Mark	Average marks the pupils have obtained within the customer month.	Non-negative real number from $[0, 100]$
Fail rate	The proportions of fail outcomes obtained on all exercises/tests within the customer month.	Non-negative real number from $[0, 1]$
Assess rate	The proportions of assessable exercises/tests taken within the customer month. Whizz online tutorial system provides both assessable exercises/tests and unassessable replays.	Non-negative real number from $[0, 1]$
Incomplete rate	The proportions of incomplete exercises/tests taken within the customer month. Whizz allows the pupils to leave lessons incomplete and resume to the last progress for next visit.	Non-negative real number from $[0, 1]$
Stack rate	The proportions of exercises/tests in stack depth 2 or 3 taken within the customer month. Whizz uses "stack depth" to mark lessons pupils have failed many times. The initial stack depth is 1, whose value will be added by 1 up to 3 if the pupil fails once. Pupils have to re-do failed lessons till a pass outcome is obtained to move forward.	Non-negative real number from $[0, 1]$
Age	Pupils' age in the customer month. We take into account monthly increment in age.	Non-negative real number
Age difference	The difference between Pupils' age and average math age assigned by Whizz in the customer month. Whizz assign each pupils a math age to indicate their ability age for a variety of topics.	Real number
Calendar month	The calendar month of the last day of the customer month.	Positive integer

Table 3: Feature definition.

We will restrict our study on these $m = 15$ features. Nevertheless, our model pipeline is very flexible to accommodate more features or remove redundant features.

3.3 Data Transformation

In general, learning algorithms benefit from *transformation* and *standardisation* of the data set. In particular, because our model replies on appropriate distributional assumptions for features, it is important to make sure features' empirical distribution not deviate too much from the distributional assumptions.

To illustrate the benefit of data transformation, we take the feature usage time as an example and compare the empirical distributions of raw data and transformed data in 3.1. The simplest distributional assumption is that each mixture component follows a Gaussian. We can hardly detect Gaussian component from the histogram of the raw data, while the transformed data appear more reasonably to be modelled as mixture of Gaussians.

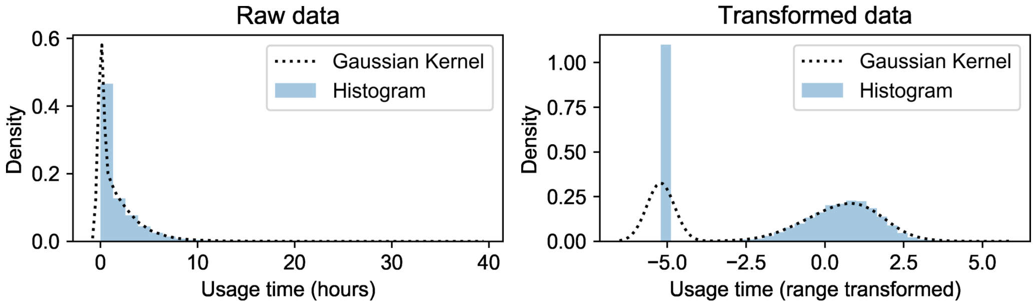

Figure 3.1: Histograms and Gaussian kernels for raw data and transformed data of feature usage time. We use Box-Cox transformation with $\lambda = 0.12$.

Standardisation of the data set is also important for our model. The motivation is to scale features into the same range so that the model maintain robustness to very small standard deviations of features. An example is shown in Figure 3.2.

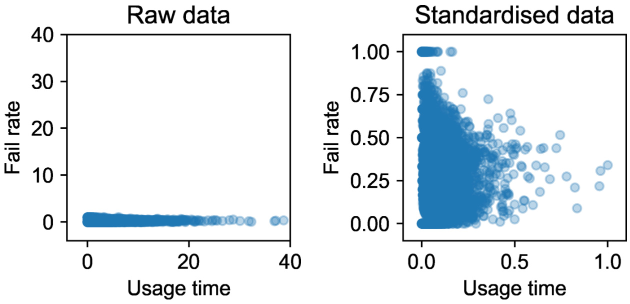

Figure 3.2: Scatter plot for data of fail rate and usage time. Usage time in hours has a range from 0 to 40 while fail rate only varies from 0 to 1. The variance of fail rate is much smaller than that of usage time. After standardisation, the features have comparable level of values.

Formally, we have employed 3 transformations in sequence to make our data more suitable for mixture model learning task. Recall that we denote the feature data by a matrix $\mathbf{X} = (x_{ij}) \in \mathbb{R}^{m\times n}$ (forget about the time subscript as in (1) for a moment), which describes n observed values for each of the m features. In addition, we denote the i-th row of $\mathbf{X}$ by $\mathbf{x}_{i,\mathrm{R}} = [x_{i1}\ x_{i2}\ \cdots\ x_{in}]$. The transformations are performed for each feature separately, resulting in different sets of transformation parameters for different features. To be specific, for i-th feature we describe the transformations as following.

- **Linear transformation:** The linear transformation is applied either to ensure all data to be positive for eligibility of applying the following power transformation, or to adapt to distributional modelling choice. It has the format:

  $$\mathbf{x}'_{i,\mathrm{R}} = a_i\mathbf{1} + b_i\mathbf{x}_{i,\mathrm{R}}, \tag{23}$$

  where a_i and b_i are constants and $\mathbf{1} \in \mathbb{R}^{1\times n}$ is the row vector of all ones.

- **Box-Cox power transformation:** The Box-Cox power transformation is used to modify the distributional shape of a set of data to be more normally distributed so that the data appear to more closely meet the assumptions of a statistical inference procedure that is to be applied. It has the format:

  $$x'_{ij} = \begin{cases} \frac{x_{ij}^{\lambda_i}-1}{\lambda_i} & \text{if } \lambda_i \neq 0, \\ \ln x_{ij} & \text{if } \lambda_i = 0, \end{cases} \tag{24}$$

  for $j = 1,\ 2,\ \ldots, m$. In Box-Cox transformation, λ_i is estimated by maximizing the likelihood function [3].

- **Standardisation:** We choose to scale all features into range $[1, 100]$. If we denote the maximum and minimum values of observed feature i as $x_i^{\max}$ and $x_i^{\min}$ respectively, then the standardisation is a linear transformation such that,

  $$\mathbf{x}'_{i,\mathrm{R}} = \mathbf{1} + \frac{100-1}{x_i^{\max} - x_i^{\min}}\left(\mathbf{x}_{i,\mathrm{R}} - x_i^{\min}\mathbf{1}\right). \tag{25}$$

We keep track of all parameters $\{a_i, b_i, \lambda_i, x_i^{\min}, x_i^{\max}\}_{i=1}^m$ involved in the data transformation and standardisation process, because they are needed in prediction task where the feature data for new pupils will be transformed and standardised using the same parameters.

4 Clustering Analysis

Given extracted feature data, we can model features' distributions, fit DP mixtures, fit Markov chain and eventually make implications on the results (process 2-5.1 in the modelling pipeline in Table 1). Before that, we would like to make assumption on customer month independence and verify it through data.

4.1 Customer Month Independence

How does cancellation rate change over time? To see this, we group subscribers by customer month and calculate the number of each group. Suppose there are n_t pupils in customer month t for $t = 1, \dots, T$, then the cancellation rate at t is defined as n_{t-1}/n_t for $t = 2, \dots, T$. Recall from section 3.1 that $n_1 = 2672$ and $T = 49$. We plot the evolution of number of subscribers and cancellation rate in Figure 4.1.

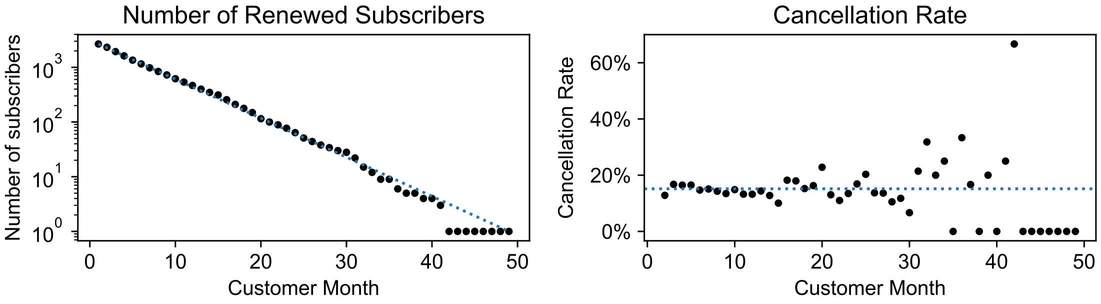

Figure 4.1: Evolution of subscriber number and cancellation rate.

The vertical axis of the subscriber number plot is log-scaled. We observe that the subscriber numbers over time fit a straight dashed line very well, inferring the number is exponentially decreasing. This is again verified by a relatively constant cancellation rate over time. There comes a lot of noises when customer month is larger than 25, because after that the number of live subscribers has reduced to less than 50 so noise dominates the estimate.

Within each customer month we can assume each customer in independent from others, then we can interpret the churn rate of the customer month group as the churn probability of each individual in that group. Hence, this observation leads to a very important assumption which greatly simplifies our model: each customer's cancellation outcome is only dependent on his features within the current customer month, not further previous ones. In other words, customer months are independent in leading to churn outcome, so we can consider all customer months together and perform a single clustering. This again implies that the Markov chain is stationary, which means $A_1 = \cdots = A_T$.

As a result, in the following analysis, we will stack feature data time series $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t, \dots, \mathbf{X}_T\}$ into a single matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ with $n = \sum_{t=1}^{T} n_t$. We compute that $n = 17861$.

4.2 Distributional Modelling of Features

We move to find appropriate density form $\phi(\cdot)$ for all $m = 15$ features. We will have a visual check of their empirical distributions and also consider their correlation structure. However, we have to firstly tackle with missing data problem introduced by the way we construct features.

4.2.1 Features with Missing Data

It is not uncommon to observe pupils who have no visits to the service at all within a customer month although they are still in subscription. This introduces missing information to features such as mark, fail rate, etc.. We cannot simply fill these information by zeros, which are indistinguishable from real zero marks obtained by other pupils and will servrely distort the distribution if their number is large.

We solve this problem by splitting pupils into three group partitions according to their activity level, where clustering task will be performed on different groups with different sets of features. The selection of features in each group ensure that there is no missing information or trivial information (for example, we can ignore “number of visit” as a feature for pupils who have no visits). We list the group definition and the associated churn rate and size in Table 4. This simple splitting can already form clusters of very different churn rates.

Group	Description	Churn rate (Size)
G1	**Inactive:** pupils having no visits at all within the customer month.	22.99% (6417)
G2	**No-assess:** pupils having visits, but only take replay mode and do not take any exercise/lesson (which will give marks, pass/fail outcome) at all within the customer month.	16.94% (425)
G3	**Complete Info:** pupils having visits, and have taken exercise/lesson within the customer month.	10.21% (11019)

Table 4: Group partitions of pupils due to activity level.

4.2.2 Independent and Multivariate Features

The simplest form for $\phi(\cdot)$ is multivariate Gaussian where each individual feature is assumed to be a mixture of normal distributions. This appears unrealistic, even after the Box-Cox transformation. We display the empirical distributions for all features in 4.2. The 4 “rate” features at the top apparently violate the Gaussian mixture assumption. The problem comes from the phenomenon called single-value-inflation, where a single value such as 0 has significantly frequent observations. In fact, mixture model is an ideal for modelling such distribution since it allows the overall distribution to be consisted of several different simpler distribution (not only Gaussian). We can model this feature separately from other features by assuming it is independent of other features.

Hence, we want to assume the 4 “rate” features to be independent from other features, and thereafter fit bespoke mixtures for each. But how likely the independence assumption holds true? We can assess the independence by at least look at the correlation structure of all features, as shown in Figure 4.3. Incomplete rate and assess rate are very negatively correlated, but uncorrelated with other features. Fail rate and stack rate have slightly negative correlation with mark, and uncorrelated with others. Overall it seems valid to model these 4 features separately from other features, and all the other 11 features will be modelled using the multivariate Gaussian mixtures (where the dependence structure will be preserved).

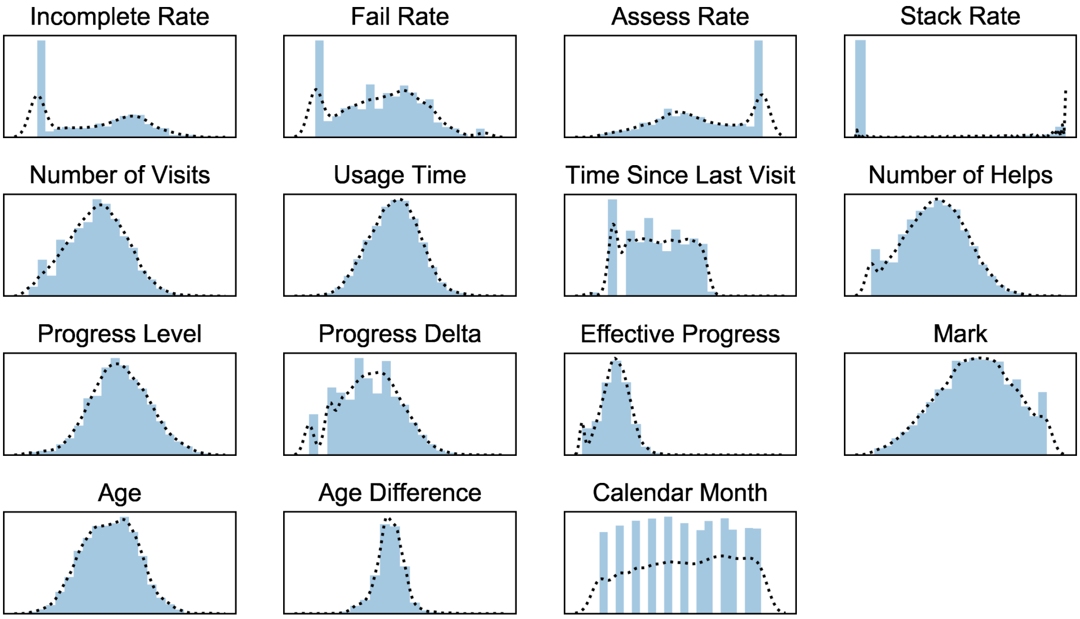

Figure 4.2: Empirical distribution (histogram) and Gaussian kernel for all features.

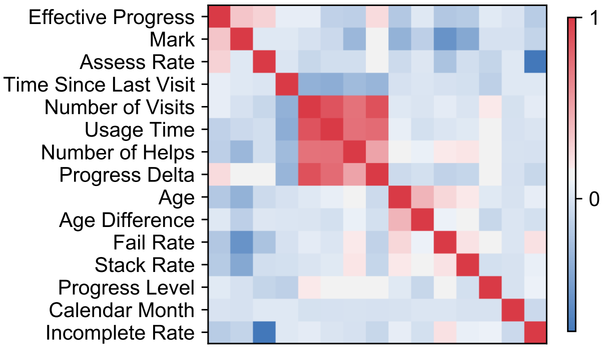

Figure 4.3: Correlation of all features. Red stands for perfect positive correlation, blue for perfect negative correlation and white for uncorrelation.

To choose the suitable mixture form for each independent feature, we have a visual check on its empirical distribution and decides the type of simple distribution and the number of components. The fitting result from EM-algorithm is visualised in Figure 4.4.

The choice of component distributions and their quantities are listed in Table 5. Note that for independent features we do not infer the number of clusters from data following Baysian approach. The reason is exactly why we assume their independence from other features and model them separately: they have idiosyncratic distributions so we want to have special treatment to capture them. The side effect is the component choice might be sensitive to observed data, and anytime when new data come in we need to verify that the choice is not severely violated. Otherwise we need alter to a

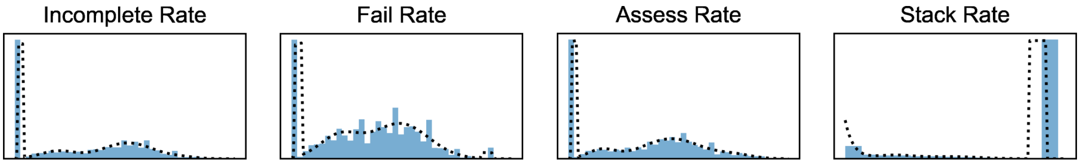

Figure 4.4: Fitting bespoke mixture model for independent features. Assess rate and stack rate data are reversed by their order to have a better fitting to simple distribution. This is achieved by linear transformation as described in (23). We can see how this outperforms compared with the Gaussian kernel for these features in Figure 4.2.

more suitable component choice. Fortunately, the component choice listed in Table 5 works well for both G2 and G3 data (G1 has none of the "rate" features defined by construction). For each component, EM-algorithm will estimate (1) the parameters of each component distribution (λ for exponential, (μ, σ) for Gaussian, etc.) and (2) the weights of each component by (locally) maximising the likelihood.

Feature	**Component distribution (Number)**
Incomplete rate	Uniform (1), Gaussian (3)
Fail rate	Uniform (2), Gaussian (2)
Assess rate	Uniform (1), Gaussian (3)
Stack rate	Exponential (1), Uniform (1), Gaussian (2)

Table 5: Component distributions and their quantity in mixture models for each independent feature.

In summary, we split the set of $m = 15$ features, by idiosyncratic attributes exhibited in their empirical distribution, into a set of $m_{\mathcal{I}} = 4$ independent features and $m_{\mathcal{M}} = 11$ multivariate Gaussian features. Let's denote the density forms for independent features as $\phi_{\mathcal{I}_i}(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ for $i = 1, \ldots, m_{\mathcal{I}}$, and the one for multivariate features as $\phi_{\mathcal{M}}(\cdot) : \mathbb{R}^{m_{\mathcal{M}}} \mapsto \mathbb{R}$. Hence, the aggregated density form is

$$\phi(\cdot) = \prod_{i=1}^{m_{\mathcal{I}}} \phi_{\mathcal{I}_i}(\cdot)\phi_{\mathcal{M}}(\cdot), \tag{26}$$

where the multivariate density is a mixture of multivariate Gaussian parametrised by weights $\boldsymbol{\pi}$ and mean-variance pair $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

4.3 Fitting Dirichelet Process Mixtures

We then fit the $m_{\mathcal{M}}$ multivariate features by Dirichelet process mixtures, which will give estimates of weights $\boldsymbol{\pi}$ and posterior distribution of the Gaussian mean-variance pair $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

A prior distribution for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, which is previously denoted as G_0 the base measure in DP, has to be pre-defined. In addition, the DP weight concentration parameter α, and the maximum number of clusters $K_{\max}$ need be specified. These are a part of model parameter choice. We list our choice in Table 6.

Parameter	Value
α	$1/m_{\mathcal{M}}$
Prior of $\boldsymbol{\mu}$	Gaussian(mean = sample mean, covariance = identity matrix)
Prior of $\boldsymbol{\Gamma}$	Wishart(degree of freedom = $m_{\mathcal{M}}$, covariance = sample covariance)
$K_{\max}$	30

Table 6: Parameter choice for DP mixture model.

We use the variational inference presented by Blei and Jordan [2]. An example result is shown in Figure 4.5. Though $K_{\max} = 30$ are pre-defined as a upper boundary for number of components, the feature data only infer 4 clusters. We count the size of each cluster and compute the churn rate for each cluster by formula (9).

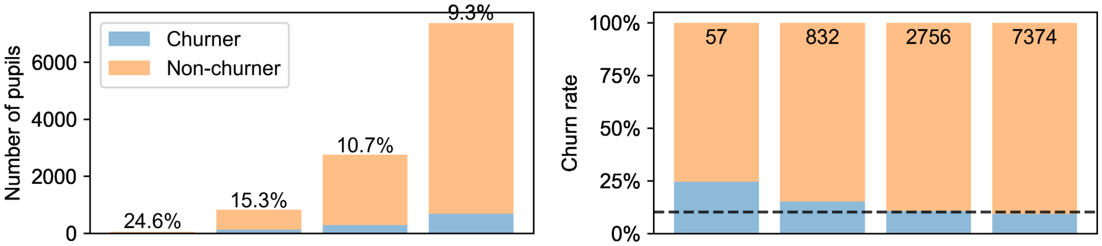

Figure 4.5: Identified clusters by multivariate features in G3, their sizes and associated churn rate. The clustering result is a local optimism which is dependent on the initialisation fed into the numerical procedure. Here we show one instance. The two plots show exactly the same thing but with different scales of vertical axis. Cluster sizes can be easily compared in left plot while it is easier to compare cluster churn rate in right plot. The dashed line in right plot indicates the group-level churn rate for G3.

Both cluster size and churn rate show sensible patterns and have meaningful implications:

- The purely behaviour-based model can infer clusters with very different churn rate. The highest cluster churn rate is 2.6 times higher than the lowest one, and 2.2 times higher than the group average.
- It is more challenging to identify pupils of higher churn probability, reflecting by the observation that the higher rate, the smaller cluster is found.

We need combine the clusters from independent and multivariate features together. By our component distributions configuration in Table 5, modelling each independent feature can find 4 clusters. Plus the 4 clusters identified from the multivariate features, our model can identify up to $1024 = 4^5$ possible clusters. Nevertheless, it ends up with 268 clusters found in G3. The size and cluster churn rate are shown in Figure 4.6. The model identifies a couple of clusters with 100% and 0% churn rate. However, the cluster size is really small for those extreme high/low churn rate. This implies that it is much easier to identify “normal” customers where the churn probability is around population average, but really challenging to distinguish “risky” or “safe” customers who have very high or low chance to cancel.

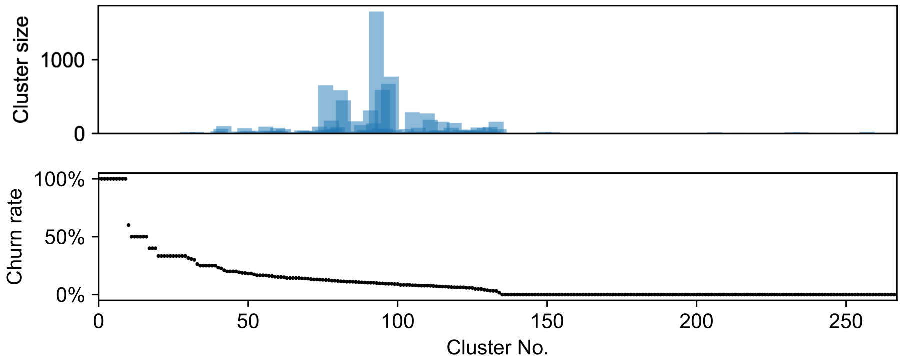

Figure 4.6: Identified 268 clusters in G3, their sizes and associated churn rate.

4.4 Assessing Churn Probability

Because we have assumed that each pupil has independent behaviours from others, and that customer month are independent in leading to churn outcome, we can interpret the cluster churn rate as the probability of churn of each individual pupil in that cluster.

We do clustering (independent features, then multivariate features and finally combine) for G1, G2 and G3 separately, and then translate the cluster churn rate into the individual churn probability. This allows us to have the distribution of individual churn probabilities for G1, G2, G3 and the whole combined sample. This is shown in Figure 4.7.

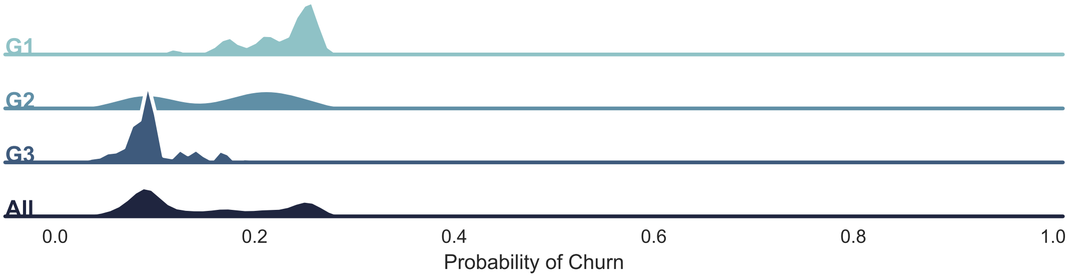

Figure 4.7: Distribution (Gaussian kernel) of individual churn probability over pupils in G1, G2, G3 and in the whole combined sample. In making this plot, our algorithm and initialisation leads to 84 clusters for G1, 28 clusters for G2 and 268 clusters for G3.

Looking at the distribution for all pupils in the sample, we can observe a few concentrations of pupils centered at churn probabilities approximately 0%, 12%, 25%. If the behaviour based model does not work at all, the inferred churn probabilities will be purely random, which will converge to a normal distribution due to central limit theorem. Distributions shown in Figure 4.7 are definitely not normal. This has shown that the behaviour based model has the predictive power to distinguish pupils with very different churn probabilities. In other words, pupils with the intention to churn exhibit different behaviours than those otherwise.

4.5 Markov States Temporal Transition Analysis

We will define the Markov states by grouping together clusters of similar level of churn rate because we want states to indicate different levels of churn risk. After forming states, we can compute transition probabilities.

4.5.1 Defining Markov States

In Figure 4.8 we show an example definition of states. We choose to form 4 states S_1, S_2, S_3 and S_4 by grouping clusters according to 3 anchor points: 0.3, 0.22, 0.35 (the three vertical dashed line shown on the left plot). The state churn rates are 0.0%, 11.7%, 25.4% and 55.6% respectively. Therefore, we can say S_1 is the safest state where pupils in this state is very unlikely to churn. In contrast, S_4 is a very risky state in the sense that pupils in this state are 5.5 times more likely than average to cancel the subscription.

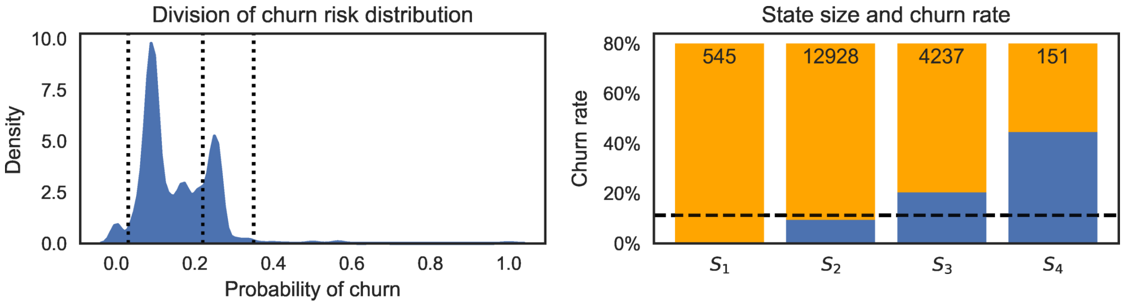

Figure 4.8: Defining Markov states by grouping together pupils of similar levels of churn risk. The right plot shows the size of each state as well as churn rate. The dashed line indicates the level of population churn rate.

The states' definition is dependent on the choice of anchor points. Whizz can choose the number and value of anchor points for forming states based on their needs. For example, if Whizz is interested in habits of very high-risk subscribers, they can increase the last anchor point.

When we want to check the churn risk for an active customer, we feed his feature data into the trained model and reach a churn probability. The predicted churn probability for a single customer might be very noisy and inaccurate due to overfitting in the clustering process. Overfitting means that the identified clusters can correspond too closely or exactly to the training data set, and therefore may fail to predict cluster for additional data point. However, when translating the churn probability into state, we are more confident about the accuracy of the predictive state. Therefore, defining state helps practical usage of the model in labeling and predicting churn risk levels.

4.5.2 Transitional Analysis

Following the state definitions in the previous section, we can move to the transitional analysis. We calculate the transition probabilities by formula (15). Note that we need to add another state S_{churn} which will only transit to itself. Hence, the complete state set is $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_{\text{churn}}\}$.

We compute the transition probability matrix as:

$$A = \begin{bmatrix} 0.11 & 0.03 & 0.01 & 0.01 & 0.00 \\ 0.66 & 0.68 & 0.33 & 0.28 & 0.00 \\ 0.23 & 0.17 & 0.40 & 0.15 & 0.00 \\ 0.00 & 0.00 & 0.01 & 0.01 & 0.00 \\ 0.00 & 0.12 & 0.25 & 0.56 & 1.00 \end{bmatrix}, \tag{27}$$

recall that $a_{pq} = \mathbb{P}(s_{t+1} = S_p | s_t = S_q)$.

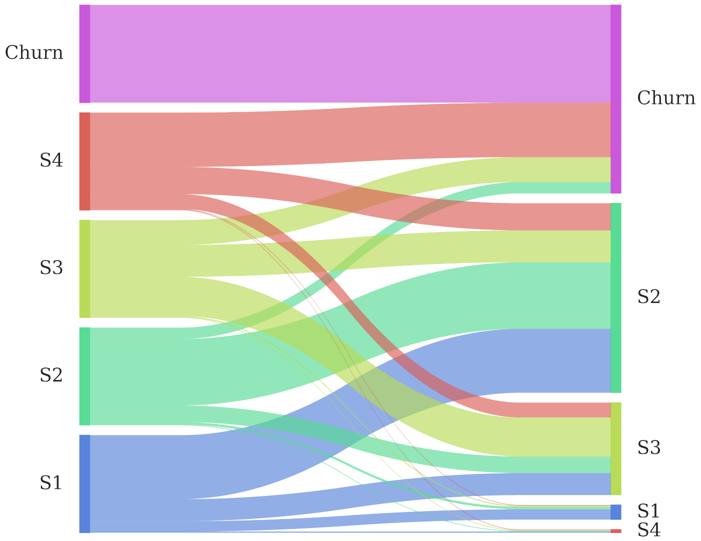

Figure 4.9: Sankey diagram showing the transition probabilities between states from time t to time $t+1$.

4.6 Feature Impact

5 Churn Probability Prediction

5.1 Prediction Workflow

5.2 Evaluating Overfitting

Imbalanced data - the reason we don't mak them balanced is because we do not want to twist numerical value of churn rate. For example, if we balance the data set perfectly, then the population churn rate will become 50%. After we do the clustering task, it is not straightforward to transform back the cluster churn rate for the balanced data set to that for the original imbalanced data set.

6 Conclusion

References

[1] D. Blackwell and J. MacQueen. Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

[2] D. M. Blei and M. I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Anal.*, 1(1):121–143, 03 2006.

[3] G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological*, pages 211–252, 1964.

[4] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[5] S. F. Slater and J. C. Narver. Intelligence generation and superior customer value. *Journal of the Academy of Marketing Science*, 28(1):120, Dec 2000.