Journal of Machine Learning Research 17 (2016) 1-51 Submitted 8/15; Revised 3/16; Published 10/16

Online Trans-dimensional von Mises-Fisher Mixture Models for User Profiles

Xiangju Qin XIANGJU.QIN@UCDCONNECT.IE
Pádraig Cunningham PADRAIG.CUNNINGHAM@UCD.IE
School of Computer Science
University College Dublin
Belfield, Dublin 4, Ireland

Michael Salter-Townshend SALTER@STATS.OX.AC.UK
Department of Statistics
University of Oxford
24-29 St Giles, Oxford, OX1 3LB, UK

Editor: David Dunson

Abstract

The proliferation of online communities has attracted much attention to modelling user behaviour in terms of social interaction, language adoption and contribution activity. Nevertheless, when applied to large-scale and cross-platform behavioural data, existing approaches generally suffer from expressiveness, scalability and generality issues. This paper proposes trans-dimensional von Mises-Fisher (TvMF) mixture models for $\mathcal{L}_2$ normalised behavioural data, which encapsulate: (1) a Bayesian framework for vMF mixtures that enables prior knowledge and information sharing among clusters, (2) an extended version of reversible jump MCMC algorithm that allows adaptive changes in the number of clusters for vMF mixtures when the model parameters are updated, and (3) an online TvMF mixture model that accommodates the dynamics of clusters for time-varying user behavioural data. We develop efficient collapsed Gibbs sampling techniques for posterior inference, which facilitates parallelism for parameter updates. Empirical results on simulated and real-world data show that the proposed TvMF mixture models can discover more interpretable and intuitive clusters than other widely-used models, such as k-means, non-negative matrix factorization (NMF), Dirichlet process Gaussian mixture models (DP-GMM), and dynamic topic models (DTM). We further evaluate the performance of proposed models in real-world applications, such as the churn prediction task, that shows the usefulness of the features generated.

Keywords: Mixture Models, von Mises-Fisher, Bayesian Nonparametric, Temporal Evolution, User Modelling

1. Introduction

Recent years have witnessed an increasing population of online peer production communities, such as *Wikipedia*, *Stack Overflow* and *OpenStreetMap*, which rely on contributions from volunteers to build knowledge, software artifacts and navigational tools, respectively. The growing popularity and importance of these communities requires a better understanding and characterisation of user behaviour so that the communities can be better managed, new services delivered, challenges and opportunities detected. For instance, by understanding the general lifecycles that users go through and the key features that distinguish different user groups and different life stages, we can develop

techniques for the following applications (Qin et al., 2014): (i) predict whether a user is likely to abandon the community; (ii) develop intelligent software to recommend tasks for users within the same life-stage. Moreover, social interaction and contribution behaviour of contributors plays a significant role in shaping the health and sustainability of online communities.

Different from text documents, which are commonly represented as term-frequency vectors, user behavioural data derived from online communities are generally represented as unit vectors. For instance, the level of linguistic change for online users in beer rating websites is denoted as a numeric feature (Danescu-Niculescu-Mizil et al., 2013). The measures used to quantify the centrality of members' positions in social networks are naturally numeric measurements (Rowe, 2013; Chan et al., 2010). The quality of questions, answers, and comments posted by users on Q&A sites are also numeric measures (Furtado et al., 2013). Existing approaches to identify patterns of user behaviour include principle component analysis, clustering analysis and entropy-based methods. However, these studies tend to be application-specific and suffer from scalability and generality issues due to the constrained feature set and the inherent limitations of the approaches employed. Additionally, the existing approaches fail to capture a mixture of user interests over time.

On the other hand, over the last decade, there have been significant advances in topic models which develop automatic text analysis techniques to discover latent structures from time-varying document collections (e.g. Blei and Lafferty (2006); Ahmed and Xing (2010); Gopal and Yang (2014)) and from time-varying user activity data in computational advertising (Ahmed et al., 2011). Topic models generally work well in document collections and user activity data where the data are represented in term-frequency format. However, many traditional topic models are not applicable to scenarios where the data are represented as unit vectors, e.g. the term frequency-inverse document frequency (tf-idf) representation of documents. Based on von Mises-Fisher (vMF) distributions, Gopal and Yang (2014) proposed dynamic clustering models which combine the success of normalised representation and flexibility of graphical models, and found that their proposed models can discover more intuitive clusters than existing approaches. Nevertheless, the vMF clustering models by Gopal and Yang (2014) did not take into consideration the dynamic evolution in the number of clusters and the birth/death of clusters over time. This work makes the following contributions:

- Extend the reversible jump Markov Chain Monte Carlo (RJMCMC) algorithm (Richardson and Green, 1997) for directional distribution (i.e. vMF mixtures).
- Enhance the Bayesian and temporal vMF mixture models by Gopal and Yang (2014) by integrating with our extended version of RJMCMC algorithm, which empowers both models with the ability to change the number of clusters automatically and to refine an inappropriate initialization of model parameters.
- Apply the model to analyse time-varying user behaviour data in online communities, in particular Wikipedia. Compared with previous works in this direction (Danescu-Niculescu-Mizil et al., 2013; Rowe, 2013; Furtado et al., 2013; Chan et al., 2010), the proposed model is more general and can be applied to model user behaviour in online communities (e.g. *Wikipedia*, *Stack Overflow*, *Twitter*, and *Facebook*) whenever user behavioural data are available.

We develop efficient collapsed Gibbs sampling techniques for the proposed models, which allows parallelism for parameter updates. The empirical comparison on synthetic and real-world data demonstrates that the proposed model can generate a more intuitive and interpretable clustering than other popular tools, such as k-means (Hartigan and Wong, 1979), non-negative matrix factorization

(NMF) (Lee and Seung, 1999), Dirichlet process Gaussian mixture models (DP-GMM) (Chang and Fisher III, 2013), and dynamic topic models (DTM) (Blei and Lafferty, 2006).

The rest of the paper is organised as follows. Section 2 provides an overview of related work, followed by an introduction to the von Mises-Fisher (vMF) distribution in Section 3. In Section 4, we present posterior inference for Bayesian von Mises-Fisher mixture models using collapsed Gibbs sampling techniques, model exploration using our extension of the reversible jump MCMC algorithm, and an online trans-dimensional von Mises-Fisher mixture model for time-varying user behavioural data. We demonstrate the empirical performance of the proposed models using synthetic and real-world data in Section 5. We then present an application of the features generated by the models for user clustering and churn prediction tasks, followed by discussion and concluding remarks in Section 6. The appendix includes detailed derivations for model inference and additional detailed analysis of the models.

2. Related Work

In this section, we provide an overview of the main lines of research underpinning this work, and discuss how our work leverages and advances the state-of-the-art techniques.

2.1 Modelling User Behaviour

Recently, researchers have approached the issue of modelling online user behaviour from different perspectives. They have so far focused on a separate set or combination of user properties, such as information exchange behaviour in discussion forums (Chan et al., 2010), social and/or lexical dynamics in online platforms (Danescu-Niculescu-Mizil et al., 2013; Rowe, 2013), and diversity of contribution behaviour on Q&A sites (Furtado et al., 2013). These studies generally employed either principle component analysis and clustering analysis to identify user profiles (Chan et al., 2010; Furtado et al., 2013) or entropy measures to track social and/or linguistic changes throughout user lifecycles (Danescu-Niculescu-Mizil et al., 2013; Rowe, 2013). While previous studies provide insights into community composition, user profiles and their dynamics, they have limitations either in their definition of lifecycle periods (e.g. dividing each user's lifetime using a fixed time-slicing approach (Danescu-Niculescu-Mizil et al., 2013) or a fixed activity-slicing approach (Rowe, 2013)) or in the expressiveness of user lifecycles in terms of the evolution of expertise and user activity for users and online communities over time. Specifically, previous studies failed to capture a mixture of user interests over time. Different from previous works, Qin et al. (2014) employed topic modelling to study the evolving patterns of editor behaviour in Wikipedia. They found that a number of editor roles (e.g. Technical Experts, Social Networkers) prevail in the temporal Wikipedia editor activity data, and that the features inspired by latent space representation are beneficial for the churn prediction task. However, two major limitations exist in the topic model used by Qin et al. (2014): (1) the inability to deal with numeric behavioural data, and (2) the inability to capture the birth/death of topics over time.

2.2 Parametric and Nonparametric Temporal Models

Parametric models, such as Latent Dirichlet Allocation (LDA) by Blei et al. (2003) and non-negative matrix factorization (NMF) by Lee and Seung (1999), generally assume a fixed pre-specified number of topics a priori. This assumption inevitably involves computationally expensive model com-

parison using model selection criteria such as penalised log-likelihoods (AIC and BIC) in order to choose an appropriate number of topics for a given dataset. While log-likelihood related criteria have shown success in static settings, it is obvious that doing model selection for each time period may lead to a sub-optimal solution as it ignores the role of context in model selection, or alternatively, local optima problems in model selection at each epoch may accumulate and result in a globally suboptimal solution (Wang et al., 2007; Ahmed and Xing, 2008). Nonparametric models have been suggested in order to relax the assumption of a fixed number of topics for stationary and time-varying data.

There are two major lines of research for non-parametric models: (1) hierarchical Dirichlet process (HDP, Teh et al. (2006)), and (2) trans-dimensional (split/merge) approaches. The hierarchical Dirichlet process is a Bayesian nonparametric model that can be used to cluster groups of data with a potentially infinite number of components. In HDP based nonparametric models, each observation within a group is a draw from a Dirichlet process (DP) (or mixture model), the group-specific DPs can be linked together via another DP to ensure the sharing of mixture components between groups; the well-known clustering property of the DP can provide a nonparametric prior for the number of mixture components within each group (Teh et al., 2006). HDP has been widely used to learn recurring patterns (or "topics") from document collections (e.g. the nonparametric Topics over Time (npTOT) model by Dubey et al. (2013)) and time-varying user activity data in computational advertising (the Time-Varying User Model (TVUM) by Ahmed et al. (2011)). In contrast, trans-dimension nonparametric models adapt the number of components by using a split-merge or birth-death mechanism while preserving certain properties (e.g. the zeroth, first and second moments) of the components. One well-known example of trans-dimensional models in this direction is the reversible jump MCMC algorithm by Richardson and Green (1997). Recently, Chang and Fisher III (2013) proposed a novel parallel restricted Gibbs sampling algorithm for Dirichlet process Gaussian mixture models (DP-GMM) with sub-cluster split/merge moves, and showed that their proposed sampler is orders of magnitude faster than other exact MCMC methods. In addition, based on the small-variance limit of Bayesian nonparametric von-Mises Fisher (vMF) mixture distributions (i.e. the birth-death strategy), Straub et al. (2015a) proposed two novel flexible and efficient k-means-like clustering algorithms for directional data such as surface normals in computer vision applications.

The evolving nature of data in different scenarios, such as scientific publications, news stories, and user query logs for search engines, has motivated research about temporal models to cope with the challenges of learning coherent and interpretable clusters over time (Ahmed and Xing, 2008). A good evolutionary model should be able to accommodate the dynamics of different aspects of the evolving clusters (Ahmed and Xing, 2008; Ahmed and Xing, 2010), specifically:

- Dynamics of cluster parameters. For example, in Gaussian mixture models, the mean and covariance for a mixture of Gaussians should be able to evolve according to a given time series model, such as Kalman filter that is used in the dynamic topic model (DTM) by Blei and Lafferty (2006). One principle for choosing a time series model for cluster parameters is to guarantee the smoothness of cluster parameters over time. One common strategy for this is to draw the cluster parameters at time epoch t from the corresponding distribution at the previous time $t-1$ by leveraging the smoothness assumption over cluster parameters (Blei and Lafferty, 2006; Ahmed and Xing, 2008; Ahmed and Xing, 2010; Ahmed et al., 2011; Gopal and Yang, 2014).

- Popularity of clusters over time. The popularity of clusters can change over time due to the evolving nature of data. Existing models have relied on the rich gets richer assumption to capture the trends of clusters over time (Ahmed and Xing, 2008; Ahmed and Xing, 2010; Ahmed et al., 2011).
- Automatic change in the number of clusters over time. Non-parametric models allow the clusters to remain, die out or emerge over time (Ahmed and Xing, 2008; Ahmed and Xing, 2010; Ahmed et al., 2011; Dubey et al., 2013).

To the authors' knowledge, while many aforementioned models have focused on categorical data, only the models by Gopal and Yang (2014) are designed for $\mathcal{L}_2$ normalised data. However, their models are parametric which require using model selection criteria to choose the appropriate number of clusters. In this work, based on the smoothness assumption and the idea of reversible jump MCMC algorithm, we extend their Bayesian vMF mixture model and propose an online trans-dimensional von Mises-Fisher mixture model (OTvMFMM) for time-varying user behavioural data. The proposed model not only allows us to explore the model space for clusters, but more importantly, it can model time-varying clusters that are consistent.

2.3 Models for Directional Data

The existence of directional data in many applications has attracted much attention from researchers to build models for clustering on the hypersphere. There are three lines of research related to clustering directional data: (1) Euclidean geometry based algorithms, which ignore the geometric properties of the data and usually use Euclidean distance to measure similarity or distance between data points (e.g. k-means by Hartigan and Wong (1979), the Dirichlet process Gaussian mixture model (DPGMM) by Rasmussen (2000)); (2) spherical geometry based models, which consider the inherent geometry of the data and use cosine similarity to measure similarity between data points with standardized length (e.g. the von Mises-Fisher mixture model (vMFMM) by Banerjee et al. (2005), the Spherical Topic Model (SAM) for documents by Reisinger et al. (2010), the Dirichlet process vMFMM for radiation therapy data by Bangert et al. (2010), the temporal vMF mixture model (Temporal vMFMM) for time-varying document collections by Gopal and Yang (2014)); and (3) models that consider spherical geometry and anisotropic covariance of directional data, which capture the geometric properties and different variances in each dimension of the data (e.g. the Dirichlet process tangential Gaussian mixture model (DP-TGMM) by Straub et al. (2015b)).

Essentially, the vMF distribution can be considered as a variant of the multivariate Gaussian with spherical covariance on $\mathbb{S}^{D-1}$, parameterized by cosine distance rather than Euclidean distance (Reisinger et al., 2010). Cosine distance belongs to the normalized correlation coefficient and takes into consideration the directions of the $\mathcal{L}_2$-normalized feature vectors when computing the similarity. Empirical studies have suggested the advantages of such type of directional measure over Euclidean distance in high-dimensional data particularly in information retrieval (Banerjee et al., 2005; Zhong and Ghosh, 2005; Reisinger et al., 2010; Gopal and Yang, 2014). The vMF distribution can capture the absence/presence of words, which the Multinomial distribution cannot. For instance, let $\theta = [1/3, 1/3, 1/3]$ be a Multinomial parameter (i.e. topic-word) vector, $d = [n_1, n_2, n_3]$ denote the number of occurrences of word w_1, w_2 and w_3 in document d. Assume we have two documents: $d_1 = [1, 1, 1]$ and $d_2 = [3, 0, 0]$. The two docments are more likely to be clustered together under Multi$(\cdot|\theta)$, whereas d_1 and d_2 have different densities under a corresponding vMF$(\cdot|\theta)$. The von Mises-Fisher mixture models have been shown to model sparse data (e.g. text) more accurately

than their Multinomial counterparts (Banerjee et al., 2005; Zhong and Ghosh, 2005; Reisinger et al., 2010; Gopal and Yang, 2014).

In this work, we employ the von Mises-Fisher distribution to deal with $\mathcal{L}_2$ normalised user behavioural data. Table 1 compares the capabilities of the proposed model and previous approaches. Although it is possible to make the proposed OTvMFMM aiming for anisotropic covariance using Fisher-Bingham distribution as in (Kent, 1982; Peel et al., 2001), extensions to high-dimensional data are difficult due to the normaliser of the probability density function (Straub et al., 2015b).

Table 1: Capabilities of different models

Models (Authors)	Spherical Geometry	Bayesian Inference	Anisotropic Covariance	Nonparametric	Parallelizeable	Temporal
DTM (Blei and Lafferty, 2006)	·	✓	·	·	✓	✓
npTOT (Dubey et al., 2013)	·	✓	·	✓	·	✓
TVUM (Ahmed et al., 2011)	·	✓	·	✓	✓	✓
vMFMM (Banerjee et al., 2005)	✓	·	·	·	✓	·
Temporal vMFMM (Gopal and Yang, 2014)	✓	✓	·	·	✓	✓
DP-GMM (Chang and Fisher III, 2013)	✓	✓	·	✓	✓	·
DP-TGMM (Straub et al., 2015b)	✓	✓	✓	✓	✓	·
OTvMFMM (proposed)	✓	✓	·	✓	✓	✓

Notation. In this work, random variables are denoted by capital letters (e.g. X, Y, Z), the observations of random variables (or vectors) are represented by the corresponding lower-case letters (e.g. x, y, z, μ). The set of N observations corresponding to random variable X is denoted by $\mathcal{X}=\{x_i\}_{i=1}^N$. The probability of a set of events A is denoted by $P(A)$, the probability of A given B (i.e. conditional probability) is written as $P(A|B)$. Probability density functions (for continuous random variables) and probability mass functions (for discrete random variables) are denoted by lower-case letters, e.g. $f(x)$, $f(x|\theta)$, p or q, where θ is the set of parameters for a specific distribution. $p(\cdot)$ and $q(\cdot)$ are frequently used to represent the distributions of random variables. The set of observations or prior parameters is denoted by a calligraphic letter (e.g. $\mathcal{X}$, $\mathcal{Z}$, $\mathcal{B}$). The set of real numbers is denoted by $\mathbb{R}$; the norm $\|\cdot\|$ denotes the $\mathcal{L}_2$ norm.

3. Preliminaries

This section provides a brief review of the finite von Mises-Fisher mixture models that facilitates a better understanding of the proposed models.

3.1 Von Mises-Fisher Distribution

A random D-dimensional unit vector x (i.e. $x \in \mathbb{R}^D$ and $\|x\|=1$) is said to follow the D-variate von Mises-Fisher (vMF) distribution if its probability density function is given by

$$f(x \mid \mu, \kappa) = C_D(\kappa)\exp\left(\kappa\mu^T x\right); \quad C_D(\kappa) = \frac{\kappa^{D/2-1}}{(2\pi)^{D/2} I_{D/2-1}(\kappa)} \tag{1}$$

where $\|\mu\| = 1$, $\kappa \geq 0$, $D \geq 2$; $C_D(\kappa)$ is the normalising constant, $I_{D/2-1}(\kappa)$ denotes the modified Bessel function of the first kind with order $D/2-1$ and argument κ. The concentration parameter,

κ, quantifies how tightly the distribution is concentrated around the mean direction μ. Note that $\mu^T x$ is the cosine similarity between x and μ and that κ plays the role of the inverse of variances (precision). The vMF distribution is used for clustering data on the unit hypersphere, whereas the Gaussian distribution is used for modelling data with a multivariate Normal distribution. A very useful property of the vMF distribution that we will use in this work is the preservation of the functional form under multiplication (Chiuso and Picci, 1998)

$$f(x \mid \mu_1, \kappa_1) f(x \mid \mu_2, \kappa_2) = f(x \mid \mu, \kappa), \text{ where } \mu = \frac{\kappa_1 \mu_1 + \kappa_2 \mu_2}{\|\kappa_1 \mu_1 + \kappa_2 \mu_2\|}, \ \kappa = \|\kappa_1 \mu_1 + \kappa_2 \mu_2\| \tag{2}$$

Following Mardia and Jupp (2000), the sample covariance matrix of directional data, $\mathcal{X}=\{x_i\}_{i=1}^N$, about μ is defined by:

$$S = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \tag{3}$$

which is an unbiased estimator of the covariance matrix (i.e. $\Sigma = S$).

3.2 Finite von Mises-Fisher Mixture

Let $f(x|\theta_h)$ denote a vMF distribution with parameters $\theta_h=(\mu_h, \kappa_h)$ for $h \in [1, H]$. Banerjee et al. (2005) proposed a simple vMF mixture model with the density of vMF mixtures given by

$$f(x \mid \{\pi_h, \theta_h\}_{h=1}^H) = \sum_{h=1}^{H} \pi_h f(x|\theta_h) \tag{4}$$

where $\Theta = \{\pi_h, \theta_h\}_{h=1}^H$ are the set of parameters to be estimated, π_h are the mixing proportions which are non-negative and sum to 1 (i.e. $0 \leq \pi_h \leq 1$, $\sum_h \pi_h$=1). To sample a point from this mixture distribution, we randomly choose the h-th component with probability π_h, and then sample a point x following $f(x|\theta_h)$. Let $\mathcal{X}=\{x_i\}_{i=1}^N$ be a set of N data points that are sampled independently following Eq. (4). The mixture model in Eq. (4) can be interpreted as a missing data model if we introduce a set of membership variables (a.k.a. latent/hidden variables), $\mathcal{Z}=\{z_i\}_{i=1}^N$, for the data points (Celeux et al., 2006), indicating the specific vMF distribution from which the points are sampled. Each membership variable is a H-dimensional indicator vector, denoted by $z_i=(z_{i1}, \cdots, z_{iH})$, $z_{ih} \in \{0, 1\}$, so that $z_{ih} = 1$ if and only if x_i is generated from the vMF distribution $f(\cdot|\{\pi_h, \theta_h\}_{h=1}^H)$, conditioning on z_i. The z_{ih} are assumed to be drawn independently from the following distributions

$$P(z_{ih} = 1) = \pi_h, i \in [1, N], \ h \in [1, H] \tag{5}$$

where $P(z_i) = \prod_{h=1}^H \pi_h^{z_{ih}}$. The density of the corresponding vMF mixture model with latent variables is given by (Celeux et al., 2006)

$$f(x_i, z_i \mid \{\pi_h, \theta_h\}_{h=1}^H) = P(z_i) f(x_i|z_i, \{\pi_h, \theta_h\}_{h=1}^H) \quad = \prod_{h=1}^{H} \{\pi_h f(x_i \mid \theta_h)\}^{z_{ih}} \tag{6}$$

4. Proposed Trans-dimensional vMF Mixture Models

In this section, we first describe the Bayesian von Mises-Fisher mixture model (BvMFMM) by Gopal and Yang (2014), including inference via efficient collapsed Gibbs sampling. We then present

our extension of the reversible jump MCMC algorithm for model exploration of Bayesian vMF mixtures, and introduce the proposed online trans-dimensional von Mises-Fisher mixture model for temporal user behavioural data.

4.1 Formulation of Bayesian vMF Mixture Model

The Bayesian vMF mixture model views the vMF mixture model parameters as random variables and introduces prior distributions on them. This brings advantages, such as sharing statistical strength among mean directions and flexibility for parameter estimation (Gopal and Yang, 2014). The Bayesian vMF mixture model is very similar to the Spherical Topic Model (Reisinger et al., 2010), the only difference lies in the fact that the former allows learning cluster-specific concentration parameters while the latter keeps the concentration parameters fixed. The generative process of BvMFMM proceeds as follows:

1. Draw topic proportions, $\pi \sim \text{Dirichlet}(.|\alpha)$, from a Dirichlet with hyperparameter α.
2. Draw topics, $\mu_h \sim \text{vMF}(.|\mu_0, C_0)$, on the unit hypersphere for $h \in [1, H]$.
3. Draw concentration parameters, $\kappa_h \sim \text{logNormal}(.|m, \sigma^2)$, from a log-normal distribution with mean m and variance σ^2 for $h \in [1, H]$.
4. For each $\mathcal{L}_2$ normalised data point $x_i \in \{x_i\}_{i=1}^N$:
   (a) Draw topic indicator $z_i \sim \text{Multi}(.|\pi)$.
   (b) Draw the data point $x_i \sim \text{vMF}(.|\mu_{z_i}, \kappa_{z_i})$.

where $\text{Multi}(.|\pi)$ denotes a Multinomial distribution. The plate notation[1] of BvMFMM is given in Figure 1.

$$
\begin{aligned}
\pi &\sim \text{Dirichlet}(.|\alpha) && \text{(topic proportions)} \\
\mu_h &\sim \text{vMF}(.|\mu_0, C_0) && \text{(topics, mean directions)} \\
\kappa_h &\sim \text{logNormal}(.|m, \sigma^2) && \text{(concentration parameters)} \\
z_i &\sim \text{Multi}(.|\pi) && \text{(latent cluster variables)} \\
x_i &\sim \text{vMF}(.|\mu_{z_i}, \kappa_{z_i}) && \text{(data points)} \\
h &\in [1, H], i \in [1, N]
\end{aligned}
$$

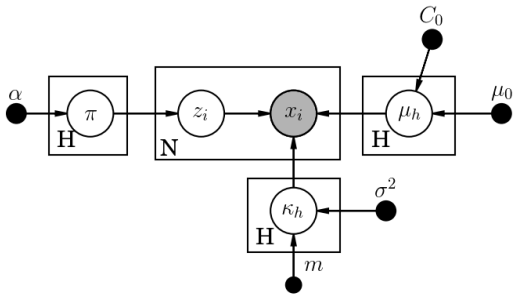


Figure 1: A graphical model representation of BvMFMM, in which nodes represent random variables, arrows denote dependency among variables, and plates denote replication. Shaded nodes correspond to observed variables, unshaded nodes represent hidden variables, and solid nodes represent the hyperparameters.

For a fully Bayesian vMF mixture model, the number of components H is considered as a random variable for which a posterior distribution should be found. Nobile (2005) suggested that compared with a uniform prior, Poisson(1) prior is strongly biased towards low H and removes

1. All the plate notations for graphical models used in this work were drawn using the Daft software provided by Dan Foreman-Mackey and David W. Hogg, available at: http://daft-pgm.org/

empty clusters. In other words, to some extent, the Poisson(1) prior acts as a penalty term that favors simpler models (i.e., ones with fewer components or factors), which is similar to the idea of AIC or BIC styled model comparison metrics (McLachlan and Peel, 2000; Gershman and Blei, 2012). Therefore, we use Poisson(1) as the prior distribution for H. From the topology of the Bayesian network, the likelihood of the complete-data, i.e., the joint distribution of all known and hidden variables that corresponds to the above generative process and the graphical model of BvMFMM in Figure 1 is given by

$$P(H, \mathcal{X}, \mathcal{Z}, \{\pi_h, \theta_h\}_{h=1}^{H} | \mathcal{B}) = P(H) f(\pi|\alpha) \prod_{h=1}^{H} f(\mu_h|\mu_0, C_0) f(\kappa_h|m, \sigma^2) \prod_{i=1}^{N} P(z_i|\pi) f(x_i|\mu_{z_i}, \kappa_{z_i}) \tag{7}$$

where $P(H)$ is the prior probability for H, $f(\pi|\alpha)$ is the prior probability for π, $f(\mu_h|\mu_0, C_0)$ and $f(\kappa_h|m, \sigma^2)$ are the prior probabilities for μ and κ respectively. $P(z_i|\pi)$ are the mixed membership priors for each data point, $f(x_i|\mu_{z_i}, \kappa_{z_i})$ are the pdfs of each observation given its mixed membership and model parameters.

In Eq. (7), the global variables are the topics μ_h, the concentration parameters κ_h and the topic proportions π, while the local variables are the per-data point topic indicators z_i. The user specified prior parameters are $\mathcal{B}$=$\{\alpha, \mu_0, C_0, m, \sigma^2\}$. We can learn these prior parameters using empirical Bayes and do not rely on the user to set any prior parameters. Alternatively, to avoid potential problems such as overfitting caused by estimating too many parameters, we can specify the values for certain prior parameters with empirical knowledge and update other parameters using empirical Bayes method. The empirical Bayes estimate for prior parameters are given in Appendix B.

Following Heinrich (2009), we can obtain the likelihood of x_i as one of its marginal distributions by integrating out the distributions π, μ and κ and summing over z_i:

$$f(x_i|m, \sigma^2, \mu_0, C_0, \alpha) = \int_{\pi} \text{Mult}(z_i|\pi) f(\pi|\alpha) \times \int_{\mu} \int_{\kappa} f(x_i|\mu_{z_i}, \kappa_{z_i}) f(\mu|\mu_0, C_0) f(\kappa|\, m, \sigma^2) \tag{8}$$

Finally, the corresponding likelihood of the complete data $\{x_i\}_{i=1}^{N}$ is given by:

$$f(\{x_i\}_{i=1}^{N}|m, \sigma^2, \mu_0, C_0, \alpha) = \prod_{i=1}^{N} f(x_i|m, \sigma^2, \mu_0, C_0, \alpha) \tag{9}$$

4.2 Inference via Collapsed Gibbs Sampling

Because the likelihood of x_i in Eq. (8) is not a closed form distribution, the exact inference of the high dimensional vMF mean parameter μ_h is generally intractable. Following Gopal and Yang (2014), we make use of the fact that the vMF distributions are conjugate and employ this fact to completely integrate out the mean parameters μ and the mixing proportions π. Therefore we need to infer only two sets of parameters $\{\mathcal{Z}, \kappa\}$. The conditional distribution for z_i is given by[2]

$$\begin{aligned} \gamma(z_{ih}) &\equiv P(z_{ih} = 1|\mathcal{Z}_{-i}, \{x_j\}_{j=1}^{N}, \kappa, m, \sigma^2, \mu_0, C_0, \alpha) \\ &\propto (\alpha + n_{h,-i}) C_D(\kappa_h) \frac{C_D(\|\kappa_h \sum_{j \neq i} z_{jh} x_j + C_0 \mu_0\|)}{C_D(\|\kappa_h (x_i + \sum_{j \neq i} z_{jh} x_j) + C_0 \mu_0\|)} \end{aligned} \tag{10}$$

2. We defer the detailed derivations to Appendix A. The $\gamma(z_{ih})$ can also be seen as the responsibility of component h for explaining the i-th data point, subject to $0 \leq \gamma(z_{ih}) \leq 1$, $\sum_h \gamma(z_{ih})$=1.

where $\mathcal{Z}_{-i}$ indicates excluding the contribution of the i-th data point. Similarly, the conditional distribution for κ_h is given by[2]

$$f(\kappa_h|\mathcal{X},\mathcal{Z},\kappa,m,\sigma^2,\mu_0,C_0) \propto \frac{C_D(\kappa_h)^{n_h}C_D(C_0)}{C_D(\|\kappa_h\sum_{j:z_{jh}=1}z_{jh}x_j+C_0\mu_0\|)}\,\text{logNormal}(\kappa_h|m,\sigma^2) \tag{11}$$

where n_h is the number of observations assigned to the h-th component, $n_h = \sum_i z_{ih}$. Since the conditional distribution for κ_h is not a closed form, we need to use a step of MCMC sampling (with appropriate distribution) to draw κ_h.

Sampling for μ and π. Finally, we need to estimate (π,μ). According to their usual definitions as directional distributions with vMF priors, applying Bayes' rule on the component z_{ih}=1 in Eq. (9) gives the full conditional posterior density function for μ as follows:

$$\begin{aligned} f(\mu_h|\{x_i\}_{i=1}^N,\{z_i\}_{i=1}^N,\mu_0,C_0) &= \frac{\prod_{z_{ih}=1}^N f(x_i|\mu_h,\kappa_h)P(\mu_h|\mu_0,C_0)}{\int\prod_{z_{ih}=1}^N f(x_i|\mu_h,\kappa_h)P(\mu_h|\mu_0,C_0)d\mu} \\ &= \frac{\prod_{z_{ih}=1}^N C_D(\kappa_h)C_D(C_0)\exp\left\{\kappa_h\mu_h^T x_i+C_0\mu_0^T\mu_h\right\}}{Z_\mu} \propto \exp\left\{\left(\kappa_h\sum z_{ih}x_i^T+C_0\mu_0^T\right)\mu_h\right\} \end{aligned} \tag{12}$$

where the updates for concentration parameter and mean direction of the posterior distribution are given by $\|\kappa_h\sum z_{ih}x_i+C_0\mu_0\|$ and $\frac{\kappa_h\sum z_{ih}x_i+C_0\mu_0}{\|\kappa_h\sum z_{ih}x_i+C_0\mu_0\|}$, respectively. The update for the mean parameter μ_h is drawn from the von Mises-Fisher distribution in Eq. (12). Similarly, according to their usual definitions as Multinomial distributions with Dirichlet[3] priors, applying Bayes' rule in Eq. (9) yields the posterior distribution for π:

$$\begin{aligned} f(\pi|\{x_i\}_{i=1}^N,\{z_i\}_{i=1}^N,\vec{\alpha}) &= \frac{\prod_{i=1}^N f(\pi|\vec{\alpha})P(z_i|\pi)}{\int\prod_{i=1}^N f(\pi|\vec{\alpha})P(z_i|\pi)d\pi} = \frac{1}{Z_\pi}\frac{1}{B(\vec{\alpha})}\prod_{h=1}^H\pi_h^{(\alpha_h-1)}\prod_{i=1}^N\prod_{h=1}^H\pi_h^{z_{ih}} \\ &= \frac{1}{Z_\pi}\frac{1}{B(\vec{\alpha})}\prod_{h=1}^H\pi_h^{(\alpha_h+n_h-1)} = Dir(\vec{\pi}|\vec{\alpha}+\vec{n_h}) \end{aligned} \tag{13}$$

Empirical Bayes estimate for prior parameters. When the user does not have enough information to specify the prior parameters, a general approach is to estimate them directly from the data. The prior parameters can be estimated by maximising the summation of the marginal likelihood of the data. The details are discussed in Appendix B.

4.3 Reversible Jump MCMC Algorithm

This section presents our extension of the reversible jump MCMC algorithm for directional distribution (i.e. vMF distribution), and then ensembles the new algorithm with the Bayesian vMF mixture model to allow adaptive change in the number of components, leading to trans-dimensional von Mises-Fisher mixture model (TvMFMM). Compared with BvMFMM, the proposed TvMFMM has the ability to explore multiple models simultaneously, which brings additional benefit - refining an inappropriate initialization of model parameters which parametric models are incapable of.

3. The Dirichlet distribution of order $H \geq 2$ with parameters $\alpha_1,\cdots,\alpha_H > 0$ is given by: $Dir(\vec{\pi}|\vec{\alpha}) = \frac{1}{B(\vec{\alpha})}\prod_{h=1}^H\pi_h^{\alpha_h-1}$, where $B(\vec{\alpha})$ is the multinomial Beta function defined as $B(\vec{\alpha}) = \frac{\prod_{h=1}^H\Gamma(\alpha_h)}{\Gamma(\sum_{h=1}^H\alpha_h)}$.

4.3.1 REVERSIBLE JUMP MOVE TYPES

Richardson and Green (1997) developed a methodology to perform Bayesian inference and model exploration for the univariate Gaussian mixture model by using the reversible jump MCMC algorithm, one sweep of which consists of six types of moves:

1. Updating the weights, π, following Eq. (13);
2. Updating the parameters, (μ, κ) , following Eq. (12) for μ and Eq. (11) for κ using MCMC sampling;
3. Updating the allocation, z, following Eq. (10);
4. Updating the hyperparameters, $\mathcal{B}$;
5. Splitting one mixture component into two, or combining two into one;
6. The birth or death of an empty component.

Moves (5) and (6) involve changing H by 1 and making necessary corresponding changes to (μ, κ, π, z), and are used for model exploration via the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970). Assume that a proposed move type t, from s=$(\mathcal{Z}, \Theta, H)$ to $\tilde{s}$=$(\mathcal{Z}', \Theta', H+1)$=$f(s, u)$, where $f(s, u)$ is an invertible deterministic function (Richardson and Green, 1997). The reverse of the move (from $\tilde{s}$ to s) can be accomplished by using the inverse transformation, so that the proposal is deterministic. The acceptance probabilities from s to $\tilde{s}$ and from $\tilde{s}$ to s are $min\{1, A\}$ and $min\{1, A^{-1}\}$ respectively, where

$$\begin{aligned} A &= \text{likelihood ratio} \times \text{prior ratio} \times \text{proposal ratio} \times \text{Jacobian} \\ &= \frac{f(\mathcal{X}|\mathcal{Z}', \Theta', \mathcal{B}, H+1)}{f(\mathcal{X}|\mathcal{Z}, \Theta, \mathcal{B}, H)} \times \frac{P(H+1)P(\mathcal{Z}', \Theta'|\mathcal{B}, H+1)}{P(H)P(\mathcal{Z}, \Theta|\mathcal{B}, H)} \times \frac{r_t(\tilde{s})}{r_t(s)q(u)} \times \left|\frac{\partial \tilde{s}}{\partial(s, u)}\right| \\ &= \frac{f(\tilde{s}|\mathcal{X})}{f(s|\mathcal{X})} \times \frac{r_t(\tilde{s})}{r_t(s)q(u)} \times \left|\frac{\partial \tilde{s}}{\partial(s, u)}\right| \end{aligned} \tag{14}$$

where $r_t(s)$ is the probability of choosing move type t when in state s, $q(u)$ is the density function of the auxiliary random variables u, the final term is the Jacobian determinant arising from the change of variables from (s, u) to $\tilde{s}$. The birth-death moves in (6) are supplements to the split-merge moves in (5) in a sense that the former are used for empty components, whereas the latter are used for non-empty components.

4.3.2 SPLIT AND MERGE MOVES

In the split-merge move in (5), the RJMCMC algorithm makes a random choice between splitting or merging existing component(s) with probabilities b_h and d_h=1-b_h respectively, depending on h. Generally, d_1=0, $b_{H_{max}}$=0, and b_h=d_h=0.5 for $h \in [2, H_{max}\text{-}1]$. The merging proposal works by choosing two random components j_1 and j_2, subject to the following constraint (adjacency condition)

$$\mu_{j_1} < \mu_{j_2}, \text{ with no other } \mu_s \in [\mu_{j_1}, \mu_{j_2}], s \neq j_1, j_2 \tag{15}$$

In the univariate setting, Richardson and Green (1997) proposed the constraints of preserving the zeroth, first and second moments of components before and after the split-merge move. In the multivariate setting, previous works (Dellaportas and Papageorgiou, 2006; Zhang et al., 2004)

generally preserved the mean vectors and covariance matrices of components via spectral decomposition of the covariance matrices. The first two moments of vMF distribution are mean direction and concentration parameter. However, a direct extension of the RJMCMC algorithm (i.e. preserving the first two moments) for vMF mixture model would be impractical for the split-merge move. Recall that the concentration parameter controls how tightly the distribution is concentrated around the mean direction, which resembles the idea behind covariance matrices - groups of similar data points would result in high (variance) values in the diagonal of the matrices. In other words, to some extent, preserving the concentration parameter is similar to preserving the statistical properties of the covariance matrices. In this work, we make use of the spectral decomposition of the covariance matrices for the split-merge moves of vMF mixtures.

Let $\Sigma_h = V_h \Lambda_h V_h^T$ be the spectral decomposition of the covariance matrix, Σ_h, of the components in Eq. (6), where Λ_h is a diagonal matrix $\Lambda_h = diag(\lambda_{h1}, \cdots, \lambda_{hD})$ with the eigenvalues of Σ_h in increasing order, and V_h is an orthogonal matrix with the eigenvectors of Σ_h in order corresponding to the eigenvalues in Λ_h. Let λ_{hd} denote the d-th largest eigenvalue of Σ_h. Let j_* be one of the H components to be considered to split, j_1, j_2 be the two proposed components, $\pi_{j_*}, \pi_{j_1}, \pi_{j_2}$ the corresponding weights, $\mu_{j_*}, \mu_{j_1}, \mu_{j_2}$ the corresponding mean vectors, $\kappa_{j_*}, \kappa_{j_1}, \kappa_{j_2}$ the corresponding concentration parameters, and $\Sigma_{j_*}, \Sigma_{j_1}, \Sigma_{j_2}$ the corresponding variance matrices. Let $u_1, u_2 = (u_{21}, \cdots, u_{2D})^T, u_3 = (u_{31}, \cdots, u_{3D})^T$ be the $2D+1$ random variables needed to construct weights, means and eigenvalues for the split move. They are generated from beta and uniform distributions

$$\begin{aligned} u_1 &\sim Beta(2,2),\ u_{21} \sim Beta(1,2D),\ u_{2d} \sim U(-1,1) \\ u_{31} &\sim Beta(1,D),\ u_{3d} \sim U(0,1),\ d \in [2,D] \end{aligned} \tag{16}$$

Let P be $D \times D$ rotation matrix with columns orthonormal unit vectors which has $D(D-1)/2$ free parameters. The elements in the lower triangular are randomly generated from uniform distribution $U(0,1)$, and the elements in other positions are determined by the fact that P is an orthonormal matrix. Then, the proposed split moves are given by

$$\begin{aligned} \pi_{j_1} &= u_1 \pi_{j_*},\ \ \pi_{j_2} = (1-u_1)\pi_{j_*} \\ \mu_{j_1} &= \mu_{j_*} - \sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}} \left(\sum_{d=1}^{D} u_{2d} \sqrt{\lambda_{j_* d}} V_{j_* d} \right) \\ \mu_{j_2} &= \mu_{j_*} + \sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}} \left(\sum_{d=1}^{D} u_{2d} \sqrt{\lambda_{j_* d}} V_{j_* d} \right) \\ \lambda_{j_1 d} &= u_{3d}(1-u_{2d}^2)\lambda_{j_* d} \frac{\pi_{j_*}}{\pi_{j_1}} \\ \lambda_{j_2 d} &= (1-u_{3d})(1-u_{2d}^2)\lambda_{j_* d} \frac{\pi_{j_*}}{\pi_{j_2}} \\ V_{j_1} &= P V_{j_*},\ \ V_{j_2} = P^T V_{j_*}, d \in [1,D] \\ \mu_{j_1} &= \frac{\mu_{j_1}}{\|\mu_{j_1}\|},\ \mu_{j_2} = \frac{\mu_{j_2}}{\|\mu_{j_2}\|} \text{ (Normalized mean directions)} \end{aligned} \tag{17}$$

It can be readily shown that these are indeed valid, with weights positive and covariance matrices positive-definite. Now we need to check whether the adjacency condition in Eq. (15) is satisfied. If the condition is satisfied, we reallocate those with $\arg\max_h \gamma(z_{ih}) = j_*$ to j_1 or j_2 using the formula

$P(z_{ih} = 1|\cdots) \propto \pi_h f(x_i|\{\pi_h, \theta_h\}_{h=1}^H)$. If the test is not passed, then the move is rejected in order to preserve the reversibility of the split/merge move.

The corresponding merge move is specified by the following expressions

$$\begin{aligned} \pi_{j_*} &= \pi_{j_1} + \pi_{j_2} \\ \pi_{j_*}\mu_{j_*} &= \pi_{j_1}\mu_{j_1} + \pi_{j_2}\mu_{j_2} \\ \pi_{j_*}[(\mu_{j_*}^T V_{j_*d})^2 + \lambda_{j_*d}] &= \pi_{j_1}[(\mu_{j_1}^T V_{j_1d})^2 + \lambda_{j_1d}] + \pi_{j_2}[(\mu_{j_2}^T V_{j_2d})^2 + \lambda_{j_2d}] \\ \mu_{j_*} &= \frac{\mu_{j_*}}{||\mu_{j_*}||} \text{ (Normalized mean directions)} \end{aligned} \tag{18}$$

The solutions of $u_1, u_2, u_3, \lambda_{j_*d}, V_{j_*}$ and P are as follows:

$$\begin{aligned} u_1 &= \pi_{j_1}/\pi_{j_*} \\ u_{2d} &= (\mu_{j_*}^T V_{j_*d} - \mu_{j_1}^T V_{j_*d})/(\sqrt{\lambda_{j_*d}\frac{\pi_{j_2}}{\pi_{j_1}}}) \\ u_{3d} &= \pi_{j_1}\lambda_{j_1d}/[\pi_{j_*}\lambda_{j_*d}(1-u_{2d}^2)] \\ \lambda_{j_*d} &= \pi_{j_*}^{-1}\left\{\pi_{j_1}[(\mu_{j_1}^T V_{j_1d})^2 + \lambda_{j_1d}] + \pi_{j_2}[(\mu_{j_2}^T V_{j_2d})^2 + \lambda_{j_2d}]\right\} - (\mu_{j_*}^T V_{j_*d})^2 \\ V_{j_*} &= \frac{1}{2}(P^T V_{j_1} + P V_{j_2}), d \in [1, D] \end{aligned} \tag{19}$$

For successful merge move, we have to reallocate those observations x_i with $\arg\max_h \gamma(z_{ih}) = j_1$ or $\arg\max_h \gamma(z_{ih}) = j_2$ to j_*. At this point, we calculate the acceptance probabilities of split and merge moves: $min\{1, A\}$ and $min\{1, A^{-1}\}$ according to Eq. (14), where

$$A = \frac{P(H+1, \mathcal{Z}', \Theta', \mathcal{B}|\mathcal{X})d_{H+1}}{P(H, \mathcal{Z}, \Theta, \mathcal{B}|\mathcal{X})b_H P_{alloc} q(u)} \times \left|det\left(\frac{\partial\Sigma}{\partial(\lambda, V)}\right)\right| \times |\,det(J)| \tag{20}$$

where P_{alloc} is the probability of making this particular allocation of data to j_1 and j_2 given by (Bouguila and Elguebaly, 2012)

$$P_{alloc} = \frac{\prod_{z_{ij_1}=1} \pi_{j_1} f(x_i|\mu_{j_1}, \kappa_{j_1}) \prod_{z_{ij_2}=1} \pi_{j_2} f(x_i|\mu_{j_2}, \kappa_{j_2})}{\prod_{z_{ij_*}=1} \pi_{j_1} f(x_i|\mu_{j_1}, \kappa_{j_1}) + \pi_{j_2} f(x_i|\mu_{j_2}, \kappa_{j_2})} \tag{21}$$

The $\frac{\partial\Sigma}{\partial(\lambda,V)}$ term is the Jacobian of the transformation from the Σ of the components to the eigenvalues and eigenvectors (Dellaportas and Papageorgiou, 2006); J is the Jacobian of the parameter transformation. The calculation of the Jacobian terms and the factorization of $P(H, \mathcal{Z}, \theta, \mathcal{B}|\{x_i\}_{i=1}^N)$ are given in Appendix C and Appendix D.

It is worth noting that it can be computationally inefficient to calculate the Jacobian term $\left|det\left(\frac{\partial\Sigma}{\partial(\lambda,V)}\right)\right|$ particularly for high-dimensional data. To solve this issue, Zhang et al. (2004) proposed a simplified multivariate Gaussian mixture model (GMM) with reversible jump MCMC algorithm, which imposed a common eigenvector matrix for the covariance matrices of all components. Their experiments showed that their proposed simplified GMM obtains good estimates on general GMMs, especially on their model exploration. In this work, we follow Zhang et al. (2004) and use a common eigenvector matrix for the covariance matrices of all the vMF components when splitting/merging the components.

Algorithm 1: Collapsed Gibbs sampling for TvMFMM

Input:
Data points: $\mathcal{X} = \{x_i\}_{i=1}^{N}$ (Unit vectors on the hypersphere)
Prior parameters: $\mathcal{B}$=$\{\alpha, \mu_0, C_0, m, \sigma^2\}$
Initial number of components: H

Output:
Estimation of model parameters

Initialise parameters: $\pi = \{\pi_h\}_{h=1}^{H}$, $\mu = \{\mu_h\}_{h=1}^{H}$, $\kappa = \{\kappa_h\}_{h=1}^{H}$;
Initialise latent variables $\mathcal{Z} = \{\{z_{ih}\}_{h=1}^{H}\}_{i=1}^{N}$ arbitrarily;
$n \leftarrow 0$;
while *NotConverged* **do**
  Sample a random variable u from Uniform(0,1): $u \leftarrow \mathrm{U}(0, 1)$;
  if $u \leqslant b_{birth\text{-}death}$ **then**
    Create or delete an empty component;
  else if $u \leqslant b_{birth\text{-}death} + b_{split\text{-}merge}$ **then**
    Split one nonempty component into two, or merge two into one;
  else
    Sample latent variables, z_{ih}, following the parallel sampling in Algorithm 2;
  end
  if *Accept the birth-death/split-merge move or Update latent variables* **then**
    Sample the weights, π_n, following Eq. (13);
    Sample the mean directions, μ_n, following Eq. (12);
    Sample the concentration parameters, κ_n, using MCMC sampling;
    Sample hyperparameters;
  end
  $n \leftarrow n + 1$;
  Check for convergence;
end
Employ the k!-means style algorithm by Celeux et al. (2000) to solve label switching problem for RJMCMC chains for mixture model estimation;

4.3.3 BIRTH AND DEATH MOVES

Our birth-death move can be adopted straightforwardly from the one used in (Richardson and Green, 1997; Zhang et al., 2004). We first make a random choice between birth or death of an empty component with the same probabilities b_k and d_k as above. For a birth, a mixing weight and parameters of the proposed component are drawn using the following distributions

$$\pi_{j_*} \sim Beta(1, H),\ \mu_{j_*} \sim \mathrm{vMF}(.|\mu_0, C_0),\ \kappa_{j_*} \sim \mathrm{logNormal}(.|m, \sigma^2) \tag{22}$$

It is necessary to rescale the existing weights in order to 'make space' for the new component using $\pi_{j'} = \pi_j(1 - \pi_{j_*})$, so that $\sum \pi_h = 1.0$. The acceptance probabilities for birth and death moves are

$min\{1, A\}$ and $min\{1, A^{-1}\}$ respectively, where

$$\begin{aligned} A &= \frac{P(H+1)}{P(H)} \times \frac{P(\pi'|H+1,\alpha)}{P(\pi|H,\alpha)} \times \frac{r_t(\tilde{s})}{r_t(s)q(u)} \times \left|\frac{\partial \tilde{s}}{\partial(s,u)}\right| \\ &= \frac{P(H+1)}{P(H)} \times \frac{\frac{\Gamma((H+1)\alpha)}{(\Gamma(\alpha))^{H+1}} \prod_{h=1}^{H+1} \pi_h'^{n_h+\alpha-1}}{\frac{\Gamma(H\alpha)}{(\Gamma(\alpha))^{H}} \prod_{h=1}^{H} \pi_h^{n_h+\alpha-1}} \times \frac{d_{H+1}}{(H_0+1)b_H} \times \frac{1}{g_{1,H}(\pi_{j_*})} \times (1-\pi_{j_*})^{H-1} \\ &= \frac{P(H+1)}{P(H)} \times \frac{1}{B(\alpha, H\alpha)} \times \pi_{j_*}^{\alpha-1} \times (1-\pi_{j_*})^{N+H\alpha-H} \times (H+1) \times \frac{d_{H+1}}{(H_0+1)b_H} \times \frac{1}{g_{1,H}(\pi_{j_*})} \times (1-\pi_{j_*})^{H-1} \end{aligned} \tag{23}$$

In Eq. (23), H_0 is the number of empty components, $g_{1,H}(\cdot)$ is the probability density function of $Beta(1, H)$ distribution.

In our implementation of the reversible jump MCMC algorithm, rather than passing through each of the six moves deterministically, following (Andrieu et al., 2003; Dellaportas and Papageorgiou, 2006), we choose to randomly select one of the three moves (i.e. Moves (3), (5) and (6)) with fixed probabilities in each iteration. We have used (.1, .4, .5) as probabilities to choose the three moves respectively. This allows some extra tuning which can potentially speed up the convergence and improve the mixing of the RJMCMC chain. The resulting collapsed Gibbs sampling procedure of TvMFMM is summarised in Algorithm 1. Note that we employ parallel sampling to update the latent variables in Algorithm 2, which is very similar to state synchronization in the parallel topic models proposed by Smola and Narayanamurthy (2010).

Algorithm 2: Parallel sampling for latent variables $\mathcal{Z}$

function `Inference`($\mathcal{X}$, $\mathcal{Z}$, α, μ_0, C_0, m, σ^2)
  Initialise $\gamma^{old}(z_{ih}) = \gamma(z_{ih})$ for all i, h;
  while *Sampling* **do**
    Read global stats $\{n_h\}_{h=1}^{H}$
    for *every component* $h \in [1, H]$ **do**
      Sample a latent, z_{ih}, conditioned on all other labels following Eq. (10).
    end
    Normalise local latent variables for data point x_i:
      $\gamma(z_{ih}) = \frac{\gamma(z_{ih})}{\sum_{j=1}^{H} \gamma(z_{ij})}$ for every $h \in [1, H]$
    Lock $\{n_h\}_{h=1}^{H}$ globally.
    Update $n_h = n_h + [\gamma(z_{ih}) - \gamma^{old}(z_{ih})]$ for every $h \in [1, H]$.
    Release $\{n_h\}_{h=1}^{H}$ globally.
  end

4.4 Online Trans-dimensional vMF Mixture Model

This section presents the procedures of collapsed Gibbs sampling for the proposed online trans-dimensional von Mises-Fisher mixture model (OTvMFMM). Given data, $\mathcal{X}=\{\{x_{t,i}\}_{i=1}^{N_t}\}_{t=1}^{T}$, where $x_{t,i} \in \mathbb{R}^D$, OTvMFMM assumes the generative model for the data given in Figure 2.

The user specified prior parameters are $\mathcal{B}=\{\alpha, \mu_{0,0}, C_0, m, \sigma^2\}$. To some extent, the prior parameter C_0 acts as a smoothing term to ensure that the parameters of the next time epoch to be similar to the previous one. Following Gopal and Yang (2014), the concentration parameters of the clusters at time t ($k_{t,h}$) are drawn from a log-Normal distribution with mean m and variance σ^2. The

$$
\begin{aligned}
\pi_t &\sim \text{Dirichlet}(.|\alpha) \\
\mu_{1,h} &\sim \text{vMF}(.|\mu_{0,0}, C_0),\ h \in [1, H] \\
\mu_{t,h} &\sim vMF(.|\mu_{t-1,h}, C_0), \text{for } t \in [2, T], h \in [1, H] \\
\mu_{t,h} &\sim \text{vMF}(.|\mu_{t,0}, C_0), t \in [2, T],\ h = H + 1 \\
k_{t,h} &\sim \text{logNormal}(.|m, \sigma^2) \\
z_{t,i}|\pi_t &\sim \text{Multi}(.|\pi_t) \\
x_{t,i} &\sim \text{vMF}(.|\mu_{z_{t,i}}, k_{z_{t,i}}) \\
\text{where } t \in [1, T],\ h \in [1, H],\ i &= 1, 2, \cdots N_t
\end{aligned}
$$

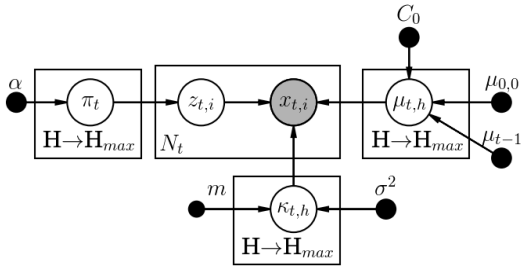


Figure 2: A graphical model representation of OTvMFMM, in which nodes represent random variables, arrows denote dependency among variables, and plates denote replication.

cluster-specific mean parameters at time t ($\mu_{t,h}$) are drawn from a vMF distribution centered around the corresponding clusters at the previous time $t-1$ or centered around $\mu_{t,0}$ with concentration C_0. This evolutionary change of the cluster parameters introduces flexibility and enables OTvMFMM to accommodate smooth changes in the mean parameter within a given cluster over time.

The inference and reversible jump MCMC algorithm for OTvMFMM can be adapted straightforwardly from those of TvMFMM introduced in Section 4.2, 4.3. The likelihood of the complete-data is given by

$$
\begin{aligned}
&P(H, \{x_{i,t}\}_{i=1}^{N_t}, \{z_{i,t}\}_{i=1}^{N_t}, \{\pi_{t,h}, \theta_{t,h}\}_{h=1}^{H}|\mathcal{B}) \\
&\quad = P(H)\ f(\pi_t|\alpha)\ P(\{z_{i,t}\}_{i=1}^{N_t}|\pi_t)\ f(\mu_t|\mu_{t,0}, C_0)\ f(\kappa_t|m, \sigma^2)\ f(\{x_{i,t}\}_{i=1}^{N_t}, \{z_{i,t}\}_{i=1}^{N_t}|\{\pi_{t,h}, \theta_{t,h}\}_{h=1}^{H}) \\
&\quad = P(H) f(\pi_t|\alpha) \prod_{h=1}^{H} f(\mu_{t,h}|\mu_{t-1,h}, C_0)\ f(\kappa_{t,h}|m, \sigma^2) \prod_{i=1}^{N_t} P(z_{t,i}|\pi)\ f(x_{t,i}|\mu_{z_{t,i}}, \kappa_{z_{t,i}})
\end{aligned}
\tag{24}
$$

The likelihood of $x_{t,i}$ can be defined as one of its marginal distributions by integrating out the distributions π_t, μ_t and κ_t and summing over $z_{t,i}$:

$$
\begin{aligned}
&f(x_{t,i}|m, \sigma^2, \mu_{t-1,h}, C_0, \alpha) = \\
&\quad \int_{\pi_t} \int_{\mu_t} \int_{\kappa_t} f(\pi_t|\alpha) \prod_{h=1}^{H} P(z_{t,i,h} = 1)\ f(x_{t,i}|\mu_{t,h}, \kappa_{t,h})\ f(\mu_{t,h}|\mu_{t-1,h}, C_0)\ f(\kappa_{t,h}|\ m, \sigma^2) \\
&\quad = \int_{\pi_t} f(\pi_t|\alpha) \prod_{h=1}^{H} P(z_{t,i,h} = 1) \int_{\mu_t} \int_{\kappa_t} \prod_{h=1}^{H} f(x_i|\mu_{t,h}, \kappa_{t,h})\ f(\mu_{t,h}|\mu_{t-1,h}, C_0)\ f(\kappa_{t,h}|\ m, \sigma^2)
\end{aligned}
\tag{25}
$$

Similarly to Section 4.2, we obtain the following updates for the posterior parameters.

$$
\begin{aligned}
\gamma(z_{t,i,h}) &\equiv P(z_{t,i,h} = 1 | \mathcal{Z}_{t,-i}, \{x_{i,t}\}_{i=1}^{N_t}, \kappa_t, m, \sigma^2, \mu_{t-1,h}, C_0, \alpha) \\
&\propto (n_{t,h} + \alpha)\, C_D(\kappa_{t,h}) \frac{C_D(\|\kappa_{t,h} \sum_{j \neq i} z_{t,j,h} x_{t,j} + C_0\, \mu_{t-1,h}\|)}{C_D(\|\kappa_{t,h}(x_{t,j} + \sum_{j \neq i} z_{t,j,h} x_{t,j}) + C_0\, \mu_{t-1,h}\|)} \\
f(\kappa_{t,h} | \mathcal{X}_t, \mathcal{Z}_t, \kappa_t, m, \sigma^2, \mu_{t-1,h}, C_0) &\propto \frac{C_D(\kappa_{t,h})^{n_{t,h}} C_D(C_0)}{C_D(\|\kappa_{t,h} \sum_j z_{t,j,h} x_{t,j} + C_0\, \mu_{t-1,h}\|)} \text{logNormal}(\kappa_{t,h} | m, \sigma^2) \\
f(\mu_{t,h} | \mathcal{X}_t, \mathcal{Z}_t, \mu_{t-1,h}, C_0) &\propto \exp\left\{ \left(\kappa_{t,h} \sum_i z_{t,i,h} x_{t,i} + C_0\, \mu_{t-1,h} \right) \mu_{t,h} \right\} \\
f(\pi_{t,h} | \{x_{i,t}\}_{i=1}^{N_t}, \{z_{i,t}\}_{i=1}^{N_t}, \alpha) &\propto (n_{t,h} + \alpha)
\end{aligned}
\tag{26}
$$

where $n_{t,h}$ is the number of observations assigned to the h-th component at time t, $n_{t,h} = \sum_i z_{t,i,h}$. The empirical updates for the prior parameters are given as follows

$$
\begin{aligned}
&\mu_{t,0} = \frac{\sum_{h=1}^{H} \mu_{t,h}}{\| \sum_{h=1}^{H} \mu_{t,h} \|},\ C_0 = \frac{\bar{r} D - \bar{r}^3}{1 - \bar{r}^2}, \text{ where}, \bar{r} = \frac{\| \sum_{h=1}^{H} \mu_{t,h} \|}{H},\ t \in [2, T] \\
&\arg\max_{\alpha > 0} -\log\left(\frac{\Gamma(H\alpha)}{\prod_{h=1}^{H} \Gamma(\alpha)} \right) + (\alpha - 1) \sum_{h=1}^{H} \pi_{t,h} \\
&m = \frac{1}{H} \sum_{h=1}^{H} \log(\kappa_{t,h}),\ \ \sigma^2 = \frac{1}{H} \sum_{h=1}^{H} \log(\kappa_{t,h})^2 - m^2
\end{aligned}
\tag{27}
$$

Similarly, the move (5) and (6) of the RJMCMC algorithm for OTvMFMM can be straightforwardly derived following the strategy for TvMFMM in Section 4.3. The acceptance probabilities of split and merge moves are $min\{1, A\}$ and $min\{1, A^{-1}\}$ respectively, where

$$
A = \frac{P(H+1, \mathcal{Z}'_t, \{\pi'_{t,h}, \theta'_{t,h}\}_{h=1}^{H}, \mathcal{B} \mid \mathcal{X}_t)}{P(H, \mathcal{Z}_t, \{\pi_{t,h}, \theta_{t,h}\}_{h=1}^{H}, \mathcal{B} \mid \mathcal{X}_t)} \times \frac{d_{H+1}}{b_H P_{alloc} q(u)} \times \left| \frac{\partial \Sigma}{\partial(\lambda, V)} \right| \times |det(J)| \tag{28}
$$

The acceptance probabilities for birth and death moves are $min\{1, A\}$ and $min\{1, A^{-1}\}$ respectively, where

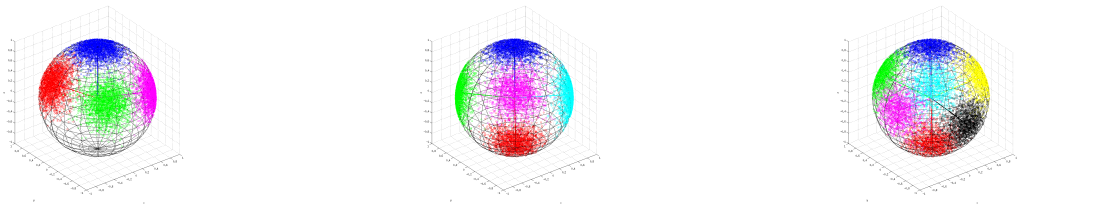
$$
\begin{aligned}
A = &\frac{P(H+1)}{P(H)} \times \frac{1}{B(\alpha, H\alpha)} \times \pi_{t,j_*}^{\alpha-1} (1 - \pi_{t,j_*})^{N + H\alpha - M} \times (H+1) \\
&\times \frac{d_{H+1}}{(H_0 + 1) b_H} \times \frac{1}{g_{1,H}(\pi_{t,j_*})} \times (1 - \pi_{t,j_*})^H
\end{aligned}
\tag{29}
$$

In Eq. (29), H_0 is the number of empty components, $g_{1,H}(\cdot)$ is the probability density function of $Beta(1, H)$ distribution.

5. Empirical Evaluation

In this section, we evaluate the proposed models on synthetic and real-world data. We used the movMF software[4] provided by Banerjee et al. (2005) to generate synthetic data with: a) 4 well-separated components; b) 5 well-separated components; c) 7 not well-separated components. Each of the synthetic datasets has a training size of 10000 and held-out test data size of 2500. The visualisation of synthetic dataset is presented in Figure 3.

4. http://suvrit.de/work/soft/movmf/



(a) 4 well-separated components (b) 5 well-separated components (c) 7 not well-separated components

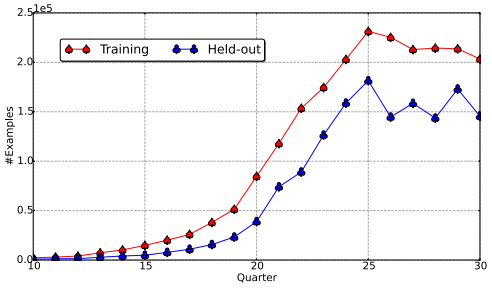
Figure 3: Visualisation of synthetic data on 3-d unit sphere.

Preprocessing of Wikipedia data. In Wikipedia, pages are subdivided into 'namespaces'[5] which represent general categories of pages based on their function. For instance, the article (or main) namespace is the most common namespace and is used to organise encyclopedia articles. In many practical applications, we might be more interested in how the actual interests of editors (in terms of the categories of Wikipedia articles they have edited) change over time. For this reason, rather than using the main namespace as one feature, we further group Wikipedia articles into clusters based on their macro-categories[6]. Because the categories for articles given by Wikipedia are generally not fine-grained, we infer the macro-categories for articles by identifying candidate categories from the DBpedia[7] category graph. DBpedia is one of the best known multi-domain knowledge bases which extracts structure information from Wikipedia Categorization system and forms a semantic graph of concepts and relations. The association between Wikipedia categories and DBpedia concepts is defined using the **subject** property of the DCIM terms vocabulary (prefixed by **dcterms:**) (Hulpus et al., 2013). A category's parent and child categories can be extracted by querying for properties **skos:broader** and **skos:broaderof**, these category-subcategory relationships create connections between DBpedia concepts. We can obtain a DBpedia category graph[8] by merging all the connections among DBpedia concepts together. With the category graph available, we can identify the macro-categories for Wikipedia articles by searching for the shortest paths from the categories associated with the articles to the macro-categories in the category graph. If multiple shortest paths exist, then the article is assigned to multiple macro-categories with weights proportional to the number of paths leading to a specific macro-category. For other complex methods of labelling topics, we recommend the readers refer to Hulpus et al. (2013).

Users can make edits to any namespace or article based on their interests and expertise. The amount of edits across all the 28 namespaces and 22 macro-categories can be considered as work archetypes. A namespace or macro-category can be considered as a 'term' in the vector space for document collections, the number of edits to that namespace/category is analogous to word frequency. A user's edit activity across different namespaces/categories in a time period can be regarded as a 'document'. The main motivation of this work is to apply topic models on the evolving user behavioural data in order to identify and characterise the patterns of change in user edit activity

5. http://en.wikipedia.org/wiki/Wikipedia:Namespace
6. At the time we collected data for this work, there were 22 macro-categories: http://en.wikipedia.org/wiki/Category:Main_topic_classifications
7. http://dbpedia.org
8. The category graph is a directed one due to the nature of category-subcategory structure.

(i.e. common work archetypes) over time in Wikipedia. For this purpose, we parsed the May 2014 dump of English Wikipedia[9], collected the edit activity of all registered users, then aggregated the edit activity of each user on a quarterly basis. In this way, we obtained a time-varying dataset consisting of the quarterly editing activity of all users from the inception of Wikipedia till May 2nd, 2014. There is an overwhelming number of users who stayed active for only one quarter. To avoid a bias towards behaviours most dominant in dataset with larger user bases, following Furtado et al. (2013), we randomly selected 20% of users who stayed active for only one quarter, included these users and those who were active for at least two quarters as our training dataset; the remaining 80% of short-term users were used as held-out dataset. The statistics and an example of the dataset are given in Figure 4. The size of the quarterly datasets range from several hundreds to about 200,000.



Observations of quarterly datasets.

An example of a simple entry in the dataset.

Uname	Quarter	Mathematics	Science	Article talk
User A	10	233	650	2
	Wikipedia	Wikipedia talk	user	user talk
	299	33	2	81

Dataset Statistics.

	#Features	#Quarters	#True clusters
Wikipedia dataset	50	21	unknown

Figure 4: Statistics and an example of Wikipedia dataset.

We implement the models in Python, and parallelize the models wherever possible by using the parallel functionality of Python. Specifically, we make use of the *multiprocessing*[10] package to implement parallelism for parameter updates, and use *Value* and *Array* data structures provided by the package to enable data sharing among multiprocessors.

Experimental settings. All our experiments were run on 32 core AMD Opteron 6134 @ 2.25Ghz with 252GB RAM. The main computational bottleneck in our collapsed Gibbs sampling algorithm is the computation of z and κ. We compared the proposed models with the following algorithms and models that are widely-used in the literature:

- K-means (Hartigan and Wong, 1979) and MiniBatchKMeans (Sculley, 2010) algorithms with random and k-means++ (Arthur and Vassilvitskii, 2007) initialisation for the centroids. Different initialisation of the centroids for k-means can affect its convergence and may lead to a local minimum. The k-means++ initialisation scheme selects initial cluster centers in a heuristic way which can speed up convergence and lead to better results than random initialisation.
- Non-negative matrix factorization (NMF) model (Lee and Seung, 1999). We used the Non-negative Double Singular Value Decomposition (NNDSVD) strategy (Boutsidis and Gallopoulos, 2008) to choose initial factors for NMF, which can produce deterministic results and avoid a poor local minimum.

9. http://dumps.wikimedia.org/enwiki/20140502/
10. https://docs.python.org/2/library/multiprocessing.html

- Dynamic topic model (DTM)[11] (Blei and Lafferty, 2006), designed for clustering of temporal document collections represented in term-frequency style format.
- Bayesian von Mises-Fisher mixture model (BvMFMM) (Gopal and Yang, 2014), designed for clustering of static numeric (and directional) data. The model is initialised with k-means++ (Arthur and Vassilvitskii, 2007) method.
- Dirichlet process Gaussian mixture model (DP-GMM) with sub-cluster split/merge moves[12] (Chang and Fisher III, 2013).

The generated synthetic data is $\mathcal{L}_2$ normalised. For Wikipedia datasets, we used the tf-idf normalised representation for NMF, BvMFMM, DP-GMM and OTvMFMM, and feature count representation (without normalisation) for DTM. For k-means, MiniBatchKMeans and NMF, we used the implementation in the scikit-learn package. If not specified, we run the models with their default parameters. If not explained specifically, TvMFMM is initialised with k-means++ strategy, and OTvMFMM is initialised with posterior estimation from the previous time point. To determine the number of clusters for parametric models (i.e. NMF, BvMFMM, and DTM) on real-world data, we experimented with different number of clusters $k \in [5, 45]$ with steps of 5 on the quarterly Wikipedia editor datasets using Non-negative Matrix Factorization (NMF) clustering, and then employed measures such as normalised pairwise mutual information (NPMI) as suggested by O'Callaghan et al. (2015) to assess model coherence for different ks. We found that overall, the run with 10 clusters generates more coherent clusters, so we set $H = 10$ for parametric models. Detailed analysis is given in Appendix F.

Chaining of clusters[13]. Label switching of the components (stemming from the posterior distribution being invariant with respect to the permutation of the component labels) is an issue which needs to be solved when analyzing MCMC samples. Briefly, label switching can be solved by imposing certain ordering constraints on component parameters. Celeux et al. (2000) proposed a k!-means style clustering algorithm to deal with label switching problem for MCMC chains, and showed the advantages and justification of k!-means clustering over loss functions for label switching problem in terms of avoiding storage of the complete MCMC chain. In this work, we employ their k!-means style algorithm for two purposes: (1) to solve label switching problem for RJMCMC chains for mixture model estimation, and (2) to chain the clusters in different quarters together in order to visualise the popularity and dynamics of clusters over time.

To evaluate the influence of the two different strategies for split-merge moves, we compared the performance of TvMF mixture model with/without common eigenvectors for split-merge moves on synthetic data. The results suggest that TvMFMM with two different strategies for the split-merge moves show similar performance and mixing properties on synthetic data. Therefore, for the results presented in the following sections, we ran TvMFMM and OTvMFMM with common eigenvectors for split-merge moves. The detailed results are provided in Appendix E.

The prior parameters for all the vMF mixtures are $\{\alpha, \mu_0, C_0, m, \sigma^2\}$. Although we provide an empirical Bayes step to estimate the priors from data, estimating too many parameters is prone to problems such as overfitting. The acceptance rate of the birth-death move in Eq. (23, 29) is sensitive to the value of α, larger value of α tends to result in lower acceptance rate for the move. We set

11. Available at: https://www.cs.princeton.edu/~blei/topicmodelling.html
12. Available at: http://people.csail.mit.edu/jchang7/code.php
13. Throughout the paper, we use the terms topic, component, cluster, mixture, and common user role interchangeably to denote the same concept.

$\alpha = 1.0$ for all models. The prior parameter μ_0 is estimated using empirical Bayes. Gopal and Yang (2014) suggested setting the prior parameter manually with relative low values (typically smaller than 10) rather than directly learning it from data. Following the suggestion, we set $C_0 = 2.0$ for all the vMF mixtures. The prior parameters m and σ^2 control the range of the concentration parameters κ, which can affect the mixing property / convergence of the RJMCMC chain. More details about this effect are discussed in Appendix H. In the following analyses (excluding those in Appendix H), we used the trace plot of the number of components and log likelihood over sweeps to assist setting appropriate values for m and σ^2.

5.1 Clustering Performance on Synthetic Data

This section compares the clustering performance of TvMF mixture model with other models on synthetic data. Following Gopal and Yang (2014), we compared the clustering performance of TvMFMM with other models on synthetic datasets using the following performance metrics:

- Adjusted Rand Index (ARI)[14].
- Normalised Mutual Information (NMI)[14].
- Adjusted Mutual Information (AMI)[14].
- Purity-related metrics[14]: Homogeneity, quantifies the extent to which each cluster contains only members of a single class; Completeness, quantifies the extent to which all members of a given class are assigned to the same cluster.

Table 2 presents the comparison of clustering performance. The results are the average of 10 runs for each method. K-means and MiniBatchKMeans algorithms are run with the true number of components and hard assignments for each dataset but with 2 different strategies to initialise the centroids. BvMFMM is also run with the true number of components. The posterior estimation of the number of components for TvMFmix are 4, 5, and 7 for the three synthetic datasets, respectively, based on the results in Appendix E. To further verify our findings, we conducted two-way significance tests using paired t-tests between TvMFMM and other models for every metric. The null hypothesis is that there is no significant difference between the performance of TvMFMM and other models.

The results show that TvMFMM performs statistically significantly better than BvMFMM on synthetic datasets with 7 not well-separated components, but both models have similar performance on datasets with 4 and 5 well-separated components; TvMFMM performs statistically significantly better than DP-GMM on all the three synthetic datasets; TvMFMM has similar Homogeneity scores as k-means and MiniBatchKMeans, but performs statistically significantly better than k-means and MiniBatchKMeans on the other four performance metrics. Note that the clustering performance of BvMFMM relies on setting the optimal number of clusters and reasonable initialisation of cluster centres, while TvMFMM solves the two issues via the use of RJMCMC algorithm. For BvMFMM, even if the number of clusters could be tuned carefully to be the optimal number for a specific dataset, a reasonable initialisation of the cluster parameters (i.e. mean directions) would still be difficult particularly for not well-separated datasets. This explains why TvMFMM performs better than BvMFMM on the synthetic dataset with 7 components. Overall, the results suggest that the proposed TvMF mixture model performs significantly better than other widely-used models such as, k-means, BvMFMM, and DP-GMM on synthetic data, i.e. points on the unit sphere.

14. http://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

Table 2: Comparison of clustering performance for different models on synthetic data. Bold face numbers indicate best performing method for the corresponding metric. The results of the paired t-test against TvMFMM are denoted by: * for significance at 5% level, ** for significance at 1% level.

True H	ARI	AMI	NMI	Homogeneity	Completeness
TvMFMM					
4	**0.925**	**0.898**	**0.899**	0.898	**0.899**
5	**0.947**	**0.941**	**0.945**	0.947	**0.944**
7	**0.809**	**0.821**	**0.829**	0.827	**0.832**
BvMFMM					
4	0.881	0.858	0.86	0.859	0.861
5	0.94	0.931	0.931	0.931	0.932
7	0.612**	0.676**	0.719**	0.677**	0.765*
DPGMM					
4	0.662**	0.652**	0.726**	0.652**	0.811**
5	0.599**	0.626**	0.727**	0.626**	0.847**
7	0.436**	0.494**	0.614**	0.495**	0.767*
Kmeans with kmeans++ initialization					
4	0.517**	0.577**	0.726**	**0.911**	0.578**
5	0.675**	0.710**	0.825**	**0.958**	0.711**
7	0.769*	0.766**	0.808*	0.853	0.767**
Kmeans with random initialization					
4	0.516**	0.576**	0.724**	0.909	0.577**
5	0.674**	0.710**	0.825**	0.958	0.711**
7	0.770*	0.766**	0.809*	**0.854**	0.767**
MiniBatchKMeans with kmeans++ initialization					
4	0.540**	0.581**	0.726**	0.906	0.582**
5	0.692**	0.714**	0.826**	0.954	0.715**
7	0.765*	0.764**	0.807*	0.85	0.765**
MiniBatchKMeans with random initialization					
4	0.544**	0.584**	0.729**	0.909	0.585**
5	0.710**	0.722**	0.833**	0.959	0.723**
7	0.775*	0.770**	0.811*	0.853	0.771**

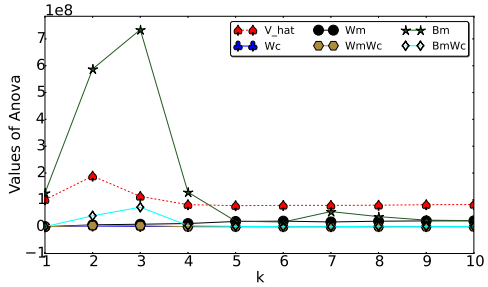
5.2 Performance on Real-world Data

In this section, we evaluate the performance of the proposed online trans-dimensional von Mises-Fishes mixture model (OTvMFMM) on Wikipedia editor activity data over 21 quarters. Because the first 9 quarters had a small number of user activity records (generally less than 1000), we choose to analyse the editor activity from the 10-th quarter to the 30-th quarter.

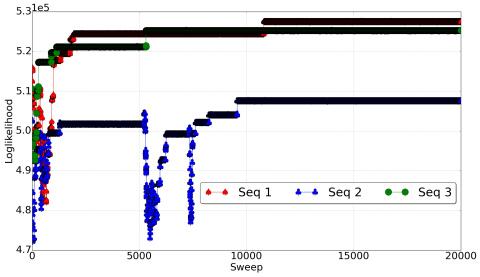
5.2.1 CONVERGENCE DIAGNOSTICS OF OTVMFMM

Before proceeding to model exploration, we diagnose the convergence of the OTvMFMMs. We follow the methodology of Brooks and Giudici (2000) which monitors particular functions of parameters (e.g. log likelihood). The method requires running I independent chains with $2T$ iterations, and then divides the I sequences into batches of length b, which gives a series of sequences of chains with length $2kb$ (where $k \in [1, T/b]$). With sequences of chains ready, we then calculate the total variations of log likelihood both between chains and between models to diagnose the convergence of the RJMCMC chain. Following Brooks and Giudici (2000), six quantities are computed:

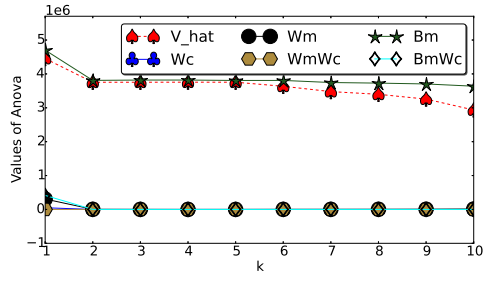
- The total variation $\hat{V}$ and the within-chain variance W_c. Essentially, the ratio $\hat{V}/W_c$ is analogous to the potential scale reduction factor (PSRF, denoted by $\hat{R}$) of Gelman and Rubin (1992).
- The within-model variance W_m, the variance within both chains and models $W_m W_c$. The comparison of W_m and $W_m W_c$, which should well approximate the true mean of within-model variance, tells us how well the chains are mixing within models.
- The between-model variance B_m, and the within-chain variation split between and averaged over models $B_m W_c$. The comparison of B_m and $B_m W_c$, which should well approximate the true between-model variance, tells us how well the chains are mixing between models.



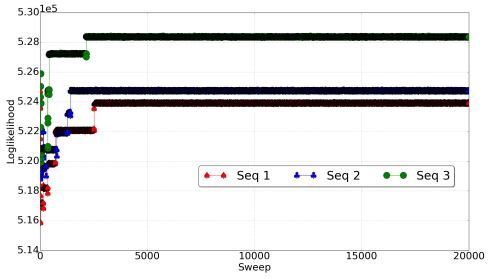
(a) k-means++ initialisation (Anova)



(b) k-means++ initialisation (Log likelihood)



(c) Posterior parameter initialisation (Anova)



(d) Posterior parameter initialisation (Log likelihood)

Figure 5: Diagnostic plots for convergence analysis and trace of the log likelihood on Wikipedia data: (a)-(b) models started with k-means++ initial means; (c)-(d) models started with the posterior mean of the previous quarter.

For details of the quantities readers can refer to Brooks and Giudici (2000). Three independent chains (each chain running 20000 iterations) were used to diagnose the convergence of OTvMFMMs started with k-means++ initial mean parameters (the initial number of components was set to 10) and started with the posterior mean of the previous quarter, respectively. We use log likelihood as the scalar parameter for convergence diagnostics which has also been used by (Brooks and Giudici, 2000; Zhong and Girolami, 2009). Figure 5 gives the diagnostic plots of convergence analysis on the 15th quarter of Wikipedia editor activity data.

The results showed that in general, OTvMFMMs started with k-means++ initial mean parameters have higher level of variations in log likelihood than those started with posterior parameter initialisation. OTvMFMMs started with k-means++ initial mean parameters became convergent after k=4, as evidenced in the corresponding trace plot for log likelihood; the overall variations of log likelihood for OTvMFMMs started with posterior mean initialisation were relatively stable throughout the range of ks, the chains were mixing very well after 3000 iterations. The results suggested that, OTvMFMM started with posterior mean initialisation converges faster than that started with k-means++ initial mean parameters. When analysing the massive temporal Wikipedia data, we notice that OTvMFMMs started with posterior mean initialisation tend to converge within 3000 iterations; we generally determine the convergence of OTvMFMM by checking the trace of the log likelihood.

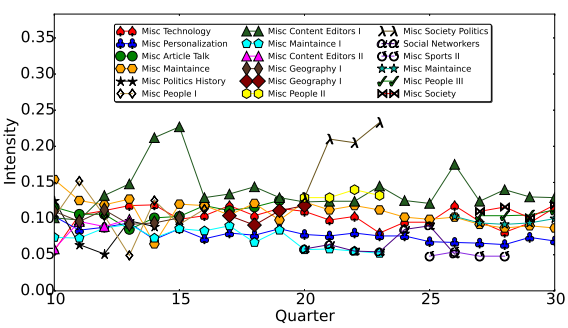
5.2.2 QUALITATIVE ANALYSIS

To show how the OTvMFMM can generate more coherent, interpretable and intuitive clusters (or common user roles) than existing models for time-varying user behavioural data, we compare the popularity of clusters over time and the evolution of top terms for selected similar clusters identified by different models. Figure 6 presents the trends of clusters over time for different models[15]. We hand-labelled the clusters generated by different models according to the top terms for each cluster. We observe that clusters generated by DTM and OTvMFMM evolve relatively smoothly in their trends over time; NMF tends to generate many clusters that appear in less than 4 quarters, indicating smoothness issues in the clusters; DTM fails to capture the birth/death of clusters over time, while OTvMFMM can capture the birth/death of clusters over time; BvMFMM generates the least smooth clusters in terms of popularity over time. Comparing the cluster labels for DP-GMM and OTvMFMM in Figure 6, we notice that DP-GMM tends to generate fewer clusters that are relatively general, whereas OTvMFMM tends to generate a larger number of clusters that are specific, interpretable and intuitive.

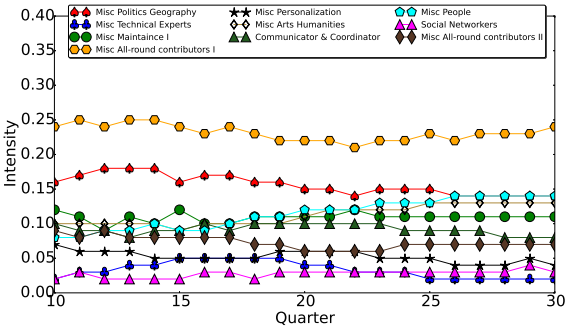
Figure 7 visualises the evolution of top terms for selected common user roles identified by different models. We observe that user roles generated by DTM and OTvMFMM generally contain a few most dominant terms, indicating interpretable and intuitive clusters; NMF also generates a few interpretable clusters; BvMFMM and DP-GMM tends to generate clusters with many dominant terms, indicating general and less interpretable clusters. The visualisation of more common user roles are provided in Figure 18 of Appendix I. Overall, the results suggest that DTM and OTvMFMM tend to generate interpretable and intuitive clusters, NMF tends to generate many clusters that appear in less than 4 quarters, BvMFMM and DP-GMM tend to generate general clusters that lack of most dominant terms.

Appendix G presents a discriminative analysis of of the topical representations by different models on the 18th quarter of Wikipedia Editor dataset.

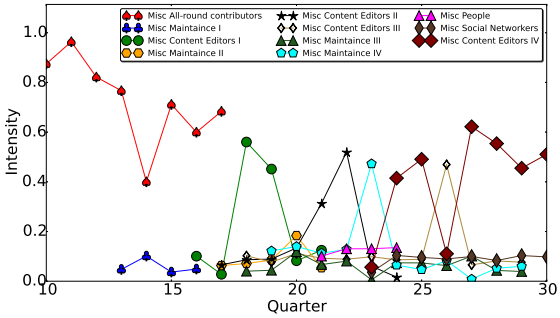
15. To improve readability, we only visualise the trends for clusters that appear at least 4 quarters.



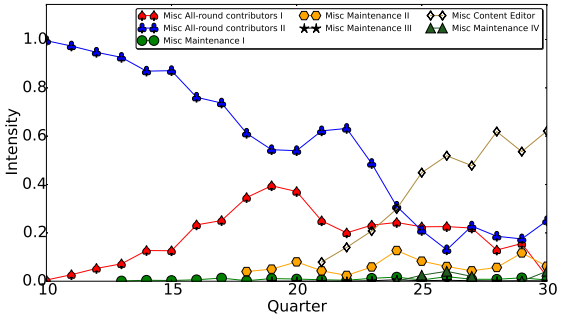
(a) NMF



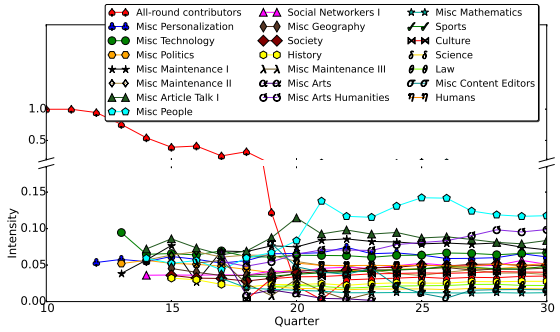
(b) DTM



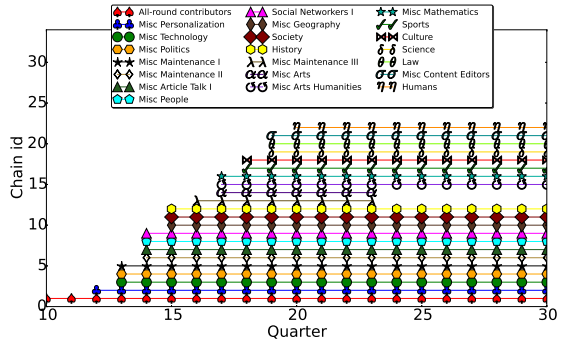
(c) BvMFMM



(d) DP-GMM

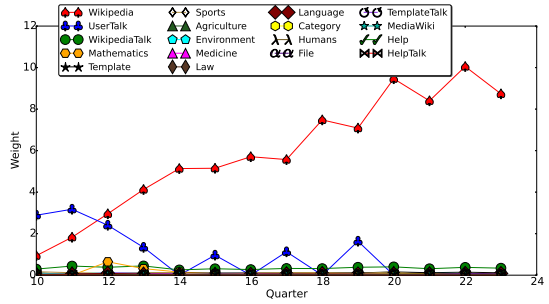


(e) OTvMFMM

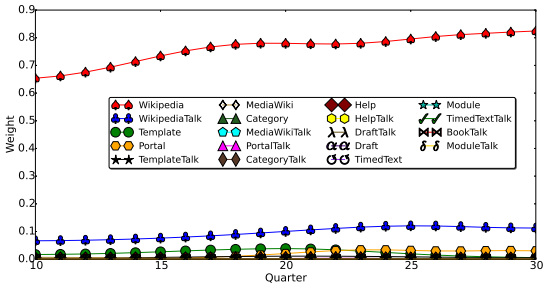


(f) Chains death-birth over time

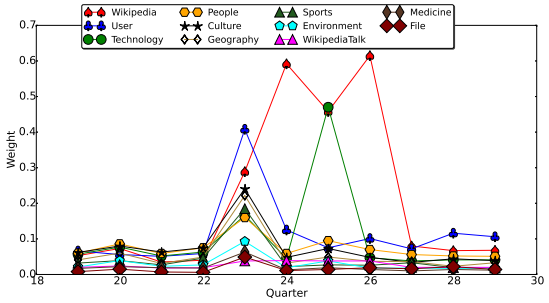
Figure 6: Popularity of common user roles over time for different models and birth-death of user roles over time for OTvMFMM. Quarter corresponds to the index of quarter. Intensity indicates the weight of user roles in each quarter, and is calculated as the percentage of user profiles assigned to that user role in a quarter. Each curve represents the trend of one user role over time. Chain id indicates the corresponding user role for OTvMFMM. NB: in (e), we make use of a discontinuity in the y-axis to better visualise the detail in the bottom of the plot.



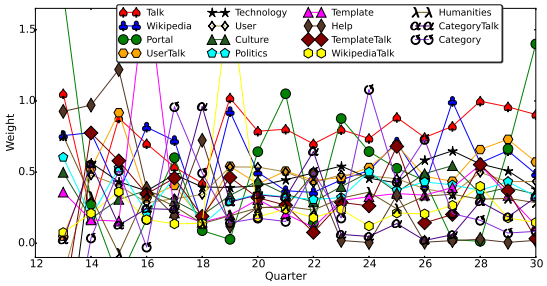
(a) NMF (Misc Maintenance I)



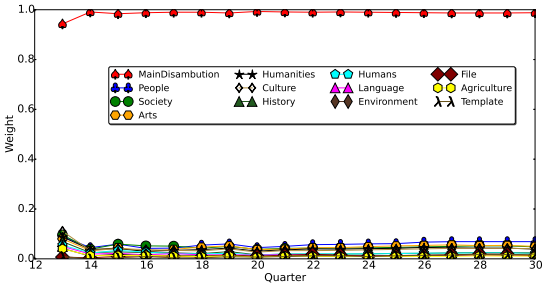
(b) DTM (Technical Experts)



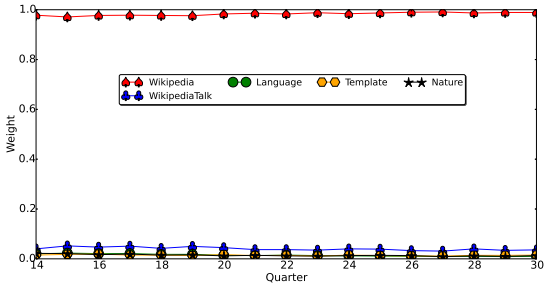
(c) BvMFMM (Misc Maintenance IV)



(d) DP-GMM (Misc Maintenance I)



(e) OTvMFMM (Misc Maintenance I)



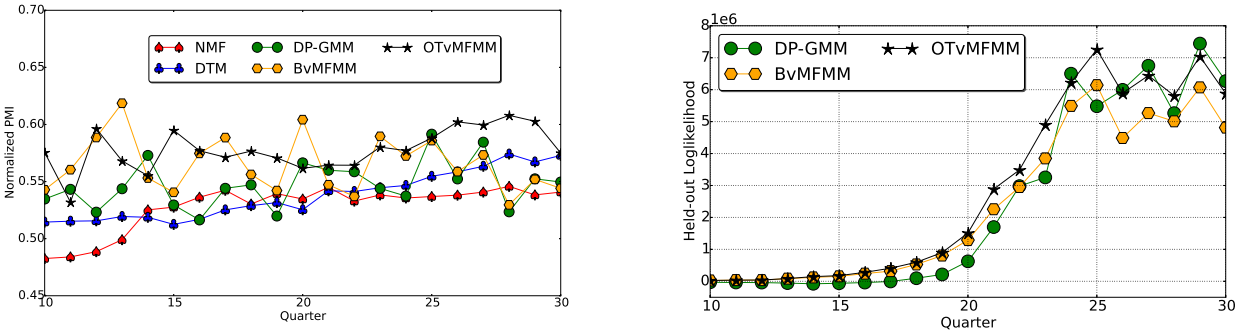
(f) OTvMFMM (Misc Maintenance II)

Figure 7: Evolution of top terms for similar common user roles (*Misc Maintenance*) identified by different models. Weight indicates the weight of features for the user roles, and is available from the model parameters.

5.2.3 QUANTITATIVE ANALYSIS

We evaluate the performance of OTvMFMM and its counterparts quantitatively from two aspects: the coherence of the clusters generated, and the perplexity (equivalently, held-out log likelihood) of the models. Traditional predictive metrics, such as perplexity, are commonly used in the literature to evaluate topic models; these metrics capture the model's predictive ability over a test set of un-

seen documents based on the parameters learned from a training set (Chang et al., 2009). Following these authors, the predictive likelihood of data point x can be approximated using $P(x \mid \mathcal{X}_{train}) = \int_{\Theta} P(x, \Theta|\mathcal{X}_{train}) \approx P(x|\hat{\Theta})\ P(\hat{\Theta}|\mathcal{X}_{train})$, where $\hat{\Theta}$ are the posterior estimation of model parameters learned from the training set. Chang et al. (2009) showed that perplexity was often negatively correlated with human judgements of topic quality, and suggested alternative measures such as topic coherence or focusing upon real-world task performance that includes human knowledge to evaluate topic quality. Topic coherence can capture the semantic interpretability of discovered topics based on their corresponding descriptor terms using measures such as normalised Pointwise Mutual Information (NPMI) (Chang et al., 2009; O'Callaghan et al., 2015). Higher coherence scores indicate better semantic interpretability, thus more coherent and interpretable topics. Figure 8 compares the coherence of clusters, and held-out log likelihood for different models[16].



(a) Cluster coherence (NPMI)

(b) Held-out log likelihood

Figure 8: Mean normalised PMI and held-out log likelihood.

We observe that the NPMI values of OTvMFMM are constantly higher than those of NMF and DTM in the range of quarters considered; the NPMI values of OTvMFMM are higher than those of BvMFMM and DP-GMM in most quarters. The NPMI values of BvMFMM and DP-GMM experience certain level of fluctuation compared with those of the other three models. The held-out log likelihood of OTvMFMM is constantly higher than that of BvMFMM in all the quarters considered; the held-out log likelihood of OTvMFMM is higher than that of DP-GMM from quarter 10 to 23, after which the measures of the two models are approximately at the same level. To summarise, the results suggest that OTvMFMM presents better predictive ability on unseen data than BvMFMM and DP-GMM, and that OTvMFMM generates more interpretable and coherent clusters than other models.

5.2.4 TIME ANALYSIS

Table 3 compares the learning time of different models on the 13th to 16th quarter of Wikipedia datasets. For DTM, the time reported was the total learning time on the 4 datasets. It is obvious from the table that NMF took the least time to learn the model, DTM took the second least time to learn the model, DP-GMM and BvMFMM took about the same amount of time to train the

16. We did not compare with the perplexity of DTM because DTM inferred model parameters using variational Bayes, and gave a lower bound of held-out log likelihood.

model. Comparing with the other four models, OTvMFMM took the maximum amount of time to learn the model on all the datasets considered. This is due to two reasons: (i) it generally takes a substantial amount of time for reversible jump MCMC styled algorithms (e.g. Richardson and Green (1997); Zhang et al. (2004); Dellaportas and Papageorgiou (2006); Zhong and Girolami (2009)) to generate convergent MCMC chains; (ii) compared with other models, updating the latent variables in OTvMFMM involves calculating the normalising constants of von Mises-Fisher distributions, which is computationally expensive, particularly for large datasets.

Table 3: Learning time in seconds for number of iterations (#Iterations) after which the algorithms converged for different models.

	NMF	DTM	DP-GMM	BvMFMM	OTvMFMM	#Obs
#Iterations	–	200	5000	40	10000	–
Quarter 13	12.65	6217.0	867.95	735.66	53333.33	7344
Quarter 14	15.83		1170.56	1503.28	69504.62	10217
Quarter 15	32.34		1722.03	1742.40	119349.18	14829
Quarter 16	27.92		2215.67	2032.45	108577.73	20129

6. Applications of User Profiles

In this section, we explore: (1) how discriminative are the generated features in distinguishing different groups of users, and (2) how useful are the features generated from patterns of change in editor activities by different models for the churn prediction task. Identifying the key features that distinguish different user groups and different life stages makes it possible to develop techniques for important applications, such as churn prediction and task recommendation. Churners present a great challenge for community management and maintenance as the turnover of established members can have a detrimental effect on the community in terms of creating communication gaps, knowledge gaps or other gaps. Qin et al. (2014) presented similar applications of user profiles. This work replicates their application scenarios in order to provide insights into the usefulness of the features generated by the proposed models in real-world applications.

6.1 Group Level Change in User Profiles

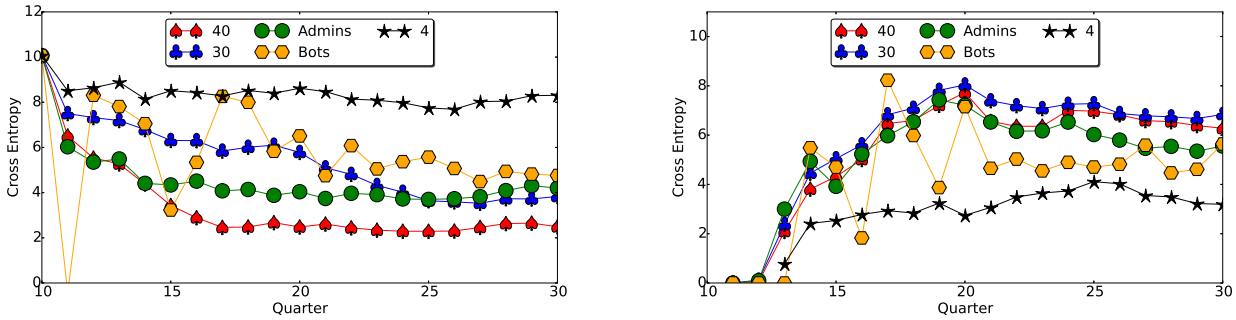
Different users are more likely to follow slightly or totally different trajectories in their lifecycle. In this analysis, we examine how different groups of users evolve throughout their lifecycle periods by comparing how each user's profile in one period is different from that in the previous periods. The historical comparison of the distribution of user profiles toward user roles is a useful indicator of how the user changes edit activity relative to past behaviour. Cross-period entropy can be used to gauge the cross-period variation in user's edit activity throughout lifecycle periods. The cross-entropy of one probability distribution P (from a given lifecycle period) with respect to another distribution Q from an earlier period (e.g. the previous quarter) is defined as follows (Rowe, 2013):

$$C(P,Q) = -\sum_{x} p(x) \log q(x) \tag{30}$$

We are interested in questions such as: what are the differences between the changes in edit activity for different groups of users (e.g. short-term vs. long-term users, admin vs. bot[17] users) over time? For this purpose, we divide the users into several groups according to the number of active quarters as in Table 4. We calculated the cross-period entropy of each user throughout their lifestages based on the distributions of their profiles toward the common user roles, and then recorded the mean of the entropy measures for each group within each period. Plotting the mean entropy values over lifecycle periods provides an overview of the general changes that each group experiences. Figure 9 visualises the cross-period entropies for different groups of editors over time.

Table 4: Groups of Editors

Name	4	30–35	⩾40	Admins	Bots
Active quarters	equal to 4	30 to 35	more than 40	(previous or current) administrators	bot users



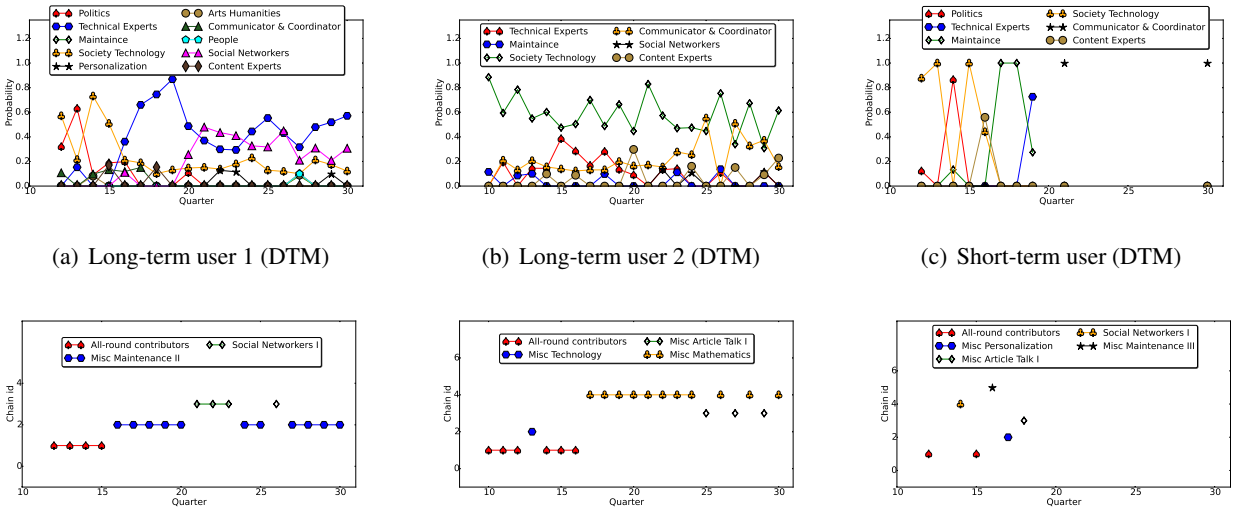
(a) DTM (b) OTvMFMM

Figure 9: Evolution of cross-period entropy for different groups of users using features generated from DTM and OTvMFMM.

We observe from Figure 9 that there is an obvious separation between the curves corresponding to the evolution of cross-period entropies of different groups of users using features generated from both models. Because DTM generates relatively general topics and tends to predict a mixture of topics (common user roles) for user profiles (a.k.a. soft clustering), whereas OTvMFMM tends to generate more specific, interpretable and intuitive topics, and then predicts a unique topic for user profiles (a.k.a. hard clustering). Special care should be taken when interpreting the trends in the evolution of cross-period entropies for the two models. For DTM, the cross-period entropies of groups with short-term users (i.e. 4) are much higher than those of groups with long-term users (i.e. 40, 30–35, and Admins), suggesting that short-term users generally experience more fluctuation in their historical edit activity and thus distribute their edit contribution among multiple namespaces and categories of articles over the course of their career. For OTvMFMM, the cross-period entropies of all groups increase over time and the cross-period entropies of groups with long-term users are much higher than those of groups with short-term users, indicating that long-term users are more

17. In Wikipedia, bots are generally programs or scripts that make repetitive automated or semi-automated edits without the necessity of human decision-making: http://en.wikipedia.org/wiki/Wikipedia:Bot_policy

likely to grow gradually in their expertise (i.e. changing from one user role to another) than short-term users. Overall, the results suggest the features generated by both DTM and OTvMFMM can capture change in editor activity over time.



(a) Long-term user 1 (DTM) (b) Long-term user 2 (DTM) (c) Short-term user (DTM)

(d) Long-term user 1 (OTvMFMM) (e) Long-term user 2 (OTvMFMM) (f) Short-term user (OTvMFMM)

Figure 10: The dynamic of profile distribution for selected long-term and short-term users for DTM (a-c) and OTvMFMM (d-f). Probability represents the probability of assigning user profile to a specific user role, and is predicted by DTM. Chain id indicates the corresponding user role for OTvMFMM.

Figure 10 provides sample individual user lifecycles for two groups of users: short-term and long-term users in terms of the evolution of profile distribution over time. We observe that there are certain level of fluctuations in the profile distributions of short-term users, signifying that short-term users generally do not develop long-term edit interests and tend to distribute their edit contribution among multiple namespaces and categories of articles over the course of their career in the community. In contrast, long-term users generally experience gradual and soft evolution in their profile distributions, and have more diversified edit preference than short-term users; these users tend to have focused/dominant long-term edit interests in one or more namespaces/categories of articles throughout their contributory lifespans. These user lifecycles provide further supports for the arguments we make in the previous paragraph.

6.2 Churn Prediction

Definition of churn prediction. Following Danescu-Niculescu-Mizil et al. (2013), we define the churn prediction task as predicting whether an editor is among the 'departed' or the 'staying' class. Considering that our dataset spans more than 5 years (i.e. 21 quarters), and that studies about churn prediction generally follow the paradigm of predicting the churn status of users in the prediction period based on user exhibited behaviour in the observation period (e.g. Weia and Chiub (2002); Danescu-Niculescu-Mizil et al. (2013)), we employed a sliding-window based method for churn

prediction. Specifically, we make predictions based on features generated from editor profile distributions in a sliding window with w=4 quarters. An editor is in the 'departed' class if she leaved the community before being active for less than m=1 quarter after the sliding window, denote the interval [w, w+m] as the departed range. Similarly, an editor is in the 'staying' class if she was active in the community long enough for a relatively large $n \geq 3$ quarters after the sliding-window, term the interval $[w + n, +\infty]$ as the staying range.

6.2.1 FEATURES FOR THE TASK

Our features are generated based on the findings reported in the previous section. For simplicity, we assume the w quarters included in the i-th sliding-window being $i = [j, \cdots, j + w - 1]$ ($j \in [10, 30]$), and denote the Probability Of Activity Profile of an editor in quarter j assigned to the k-th user role as $\text{POAP}_{i,j,k}$. We use the following features to characterise the patterns of change in editor profile distributions:

- *First active quarter*: the quarter in which an editor began edits in Wikipedia. The timestamp a user joined the community may affect her decision about whether to stay for longer.
- *Cumulative active quarters*: the total number of quarters an editor had been active in the community till the last quarter in the sliding window.
- *Fraction of active quarters in lifespan*: the proportion of quarters a user was active till the sliding window.
- *Fraction of active quarters in sliding window*: the fraction of quarters a user was active in current sliding window.
- *Similarity of profile distribution in sliding window*: quantifies the similarity of user profile distributions in any two successive quarters using cosine similarity.
- *Diversity of edit activity*: denotes the entropy of $\text{POAP}_{i,j,k}$ for each quarter j in window i. This measure captures the extent to which an editor diversified her edits toward multiple namespaces and categories of articles.
- *Cross-entropy of edit activity*: denotes the historical variation in $\text{POAP}_{i,j,k}$ compared to the same measure in previous quarters, calculated using Eq. (30). This measure captures the extent to which an editor changed her edit activity compared to her past behaviour.
- *mean* $POAP_{i,j,k}$: denotes the average of $\text{POAP}_{i,j,k}$ for each user role k in window i, and captures whether an editor focused her edits on certain namespaces and categories of articles in window i.
- $\Delta POAP_{i,j,k}$: denotes the change in $\text{POAP}_{i,j,k}$ between the quarter $j-1$ and j, measured by $\Delta\text{POAP}_{i,j,k}$=$(\text{POAP}_{i,j,k} - \text{POAP}_{i,j-1,k} + \delta)/(\text{POAP}_{i,j-1,k} + \delta)$, where δ is a small positive real number (i.e. 0.001) to avoid the case when $\text{POAP}_{i,j-1,k}$ is 0. This measure also captures the fluctuation of $\text{POAP}_{i,j,k}$ for each user role k in window i.

For each editor, the first three features are global-level features which may be updated with the sliding window, the remaining features are window-level features and are recalculated within each sliding window. The intuition behind the last four features is to approximate the evolution of editor lifecycle we sought to characterise in the previous section. The dataset is of the following form: D=(x_i, y_i), where y_i denotes the churn status of the editor, $y_i \in$ {Churner, Non-churner}; x_i denotes the feature vector for the editor.

6.2.2 TASK PERFORMANCE

Table 5: Performance of sliding-window based churn prediction using features generated from different models. The measures are averaged over all sliding windows for each model.

Models	FP Rate	Precision	Recall	F-Measure	ROC Area
DTM	0.40±0.02	0.72±0.01	0.73±0.01	0.72±0.01	0.77±0.01
OTvMFMM	0.47±0.03	0.69±0.01	0.71±0.01	0.69±0.01	0.72±0.02
NMF	0.45±0.02	0.70±0.01	0.71±0.01	0.69±0.01	0.73±0.02
BvMFMM	0.48±0.03	0.69±0.01	0.71±0.01	0.68±0.01	0.70±0.03
DP-GMM	0.45±0.04	0.70±0.01	0.71±0.01	0.69±0.02	0.72±0.04

Table 5 gives the averaged performance of sliding-window based churn prediction using features generated from different models. We observe that the best performance is obtained using features generated from DTM; churn prediction using features generated from the other four models presents similar performance. Notice that DTM and NMF generally generate a mixture of topics for user profiles, while the other three models tend to generate unique topics for user profiles. This suggests that different models may be applied to applications with different requirements. Alternatively, models such as DTM and NMF may be better at capturing change in user behaviour for churn prediction, while discriminative models such as DP-GMM and OTvMFMM may be better at identifying user expertise for task recommendation. Our results suggest that sudden changes in user behaviour can be a signal that the user is likely to abandon the community, and that features inspired by topic models are useful for churn prediction.

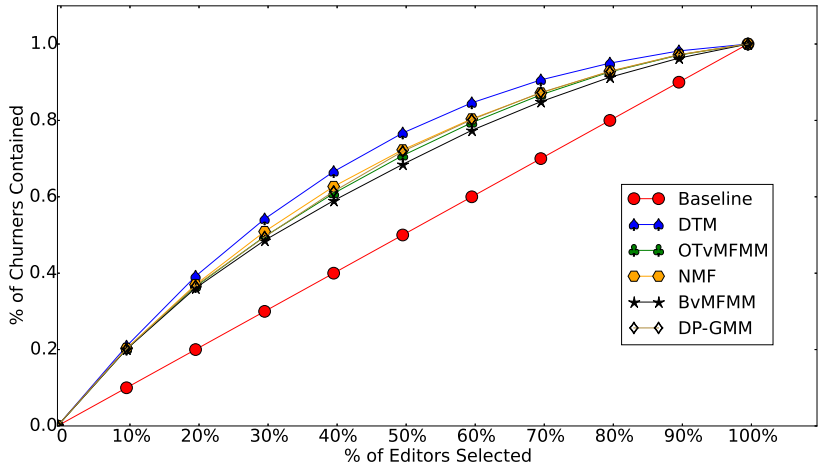


Figure 11: Lift chart of churn prediction using features generated from different models. The measures are averaged over all sliding windows.

Lift factors have been widely used by researchers to evaluate the performance of churn-prediction models (Weia and Chiub, 2002). The lift factors achieved by different models are shown in Figure 11. In a lift chart, the diagonal line represents a baseline which randomly selects a subset of editors as potential churners, i.e., it selects s% of the editors that will contain s% of the true churners, resulting in a lift factor of 1. For instance, in Figure 11, on average, all models (except Baseline) was capable of identifying 10% of editors that contained at least 20% of true churners (i.e. a lift factor of 2.0). DTM was capable of identifying 20% of editors that contained 39.3% of true churners (i.e. a lift factor of 1.966), and 30% of editors that contained 53.9% of true churners (i.e. a lift factor of 1.799). Evidently, DTM achieved slightly higher lift factors than the other four models, all models achieved higher lift factors than the baseline. Thus if the objective of the lift analysis is to identify a small subset of likely churners for an intervention that might persuade them not to churn, then this analysis suggests that all models can identify a set of 10% of users where the probability of churning is more than twice the baseline figure.

7. Conclusion

This work proposed an online trans-dimensional von Mises-Fisher mixture model (OTvMFMM) for temporal user behavioural data, which (a) enables information sharing among clusters via a Bayesian framework, (b) allows adaptive change in the number of clusters by using our extended version of the reversible jump MCMC algorithm, and (c) accommodates the dynamics of clusters for time-varying user behavioural data based on the smoothness assumption. Our efficient collapsed Gibbs sampling algorithms make the models applicable to large-scale real-world data such as Wikipedia dataset. Empirical results on synthetic and real-world data show that the proposed models can discover more interpretable and intuitive clusters than other widely-used models, such as k-means, Non-negative Matrix Factorization (NMF), Dirichlet process Gaussian mixture models (DP-GMM), and dynamic topic models (DTM). We further evaluated the performance of proposed models in real-world applications, such as churn prediction task, that shows the usefulness of the features generated.

The results show that the proposed OTvMFMM can discover more interpretable and intuitive clusters for evolving user behavioural data than DP-GMM with sub-cluster split/merge moves by Chang and Fisher III (2013), whereas the latter is found to converge much faster than the former. An interesting and promising future direction is to replace the reversible jump MCMC algorithm (Richardson and Green, 1997) with the subcluster split-merge strategy (Chang and Fisher III, 2013) in order to allow adaptive change in the number of clusters for the Bayesian von Mises-Fisher mixture models. The new integration would lead to more efficient and interpretable non-parametric models for $\mathcal{L}_2$ normalised data that combine the advantages of both OTvMFMM and DP-GMM. In addition, heterogeneous user behavioural data are ubiquitous in the sense of multiplicity of features and multiple data sources available. We would like to handle heterogeneous data and apply the models to analyse user behavioural data in other online communities.

Acknowledgements

This work was supported by Science Foundation Ireland (SFI) under Grant No. SFI/12/RC/2289 (Insight Centre for Data Analytics). Xiangju Qin was funded by University College Dublin and China Scholarship Council (UCD-CSC Joint Scholarship 2011). We thank the anonymous reviewers for their constructive suggestions.

References

Amr Ahmed and Eric P. Xing. Dynamic Non-Parametric Mixture Models and the Recurrent Chinese Restaurant Process: with Applications to Evolutionary Clustering. In *Proc. of the Eighth SIAM International Conference on Data Mining (SDM)*, pages 219–230, 2008.

Amr Ahmed and Eric P. Xing. Timeline: A Dynamic Hierarchical Dirichlet Process Model for Recovering Birth/Death and Evolution of Topics in Text Stream. In *Proc. of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 20–29, 2010.

Amr Ahmed, Yucheng Low, Mohamed Aly, Vanja Josifovski, and Alexander J. Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 114–122, 2011.

Christophe Andrieu, Freitas Nando de, Arnaud Doucet, and Michael I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1–2):5–43, 2003.

David Arthur and Sergei Vassilvitskii. k-means++: The Advantages of Careful Seeding. In *Proc. of the 18th Annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 1027–1035, 2007.

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the Unit Hypersphere using von Mises-Fisher Distributions. *Journal of Machine Learning Researh (JMLR)*, 6(1):1345–1382, 2005.

Mark Bangert, Philipp Hennig, and Uwe Oelfke. Using an Infinite Von Mises-Fisher Mixture Model to Cluster Treatment Beam Directions in External Radiation Therapy. In *Proc. of the Ninth IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 746–751, 2010.

David M. Blei and John D. Lafferty. Dynamic Topic Models. In *Proc. of the 23rd International Conference on Machine Learning (ICML)*, pages 113–120, 2006.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022, 2003.

Nizar Bouguila and Tarek Elguebaly. A fully Bayesian model based on reversible jump MCMC and finite Beta mixtures for clustering. *Expert Systems with Applications (ESA)*, 39(5):5946–5959, 2012.

Christos Boutsidis and Efstratios Gallopoulos. SVD based initialization: A head start for non-negative matrix factorization. *Pattern Recognition*, 2008.

Stephen P. Brooks and Paolo Giudici. Markov Chain Monte Carlo Convergence Assessment via Two-Way Analysis of Variance. *Journal of Computational and Graphical Statistics*, 9(2):266–285, 2000.

Richard A. Brualdi and Hans Schneider. Determinantal Identities: Gauss, Schur, Cauchy, Sylvester, Kronecker, Jacobi, Binet, Laplace, Muir, and Cayley. *Linear Algebra and its Applications*, 52/53 (1):769–791, 1983.

Gilles Celeux, Merrilee Hurn, and Christian P. Robert. Computational and Inferential Difficulties with Mixture Posterior Distributions. *Journal of the American Statistical Association (JASA)*, 95 (451):957–970, 2000.

Gilles Celeux, Florence Forbes, Christian P. Robert, and D. Mike Titterington. Deviance Information Criteria for Missing Data Models. *Bayesian Analysis*, 1(4):651–674, 2006.

Jeffrey Chan, Conor Hayes, and Elizabeth M. Daly. Decomposing Discussion Forums using User Roles. In *Proc. of the Fourth International Conference on Weblogs and Social Media (ICWSM)*, pages 215–218, 2010.

Jason Chang and John W. Fisher III. Parallel Sampling of DP Mixture Models using Sub-Cluster Splits. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, pages 620–628, 2013.

Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, pages 288–296, 2009.

Alessandro Chiuso and Giorgio Picci. Visual Tracking of Points as Estimation on the Unit Sphere. In *The confluence of vision and control*, pages 90–105. Springer Link, 1998.

Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. No Country for Old Members: User Lifecycle and Linguistic Change in Online Communities. In *Proc. of the 22nd International World Wide Web Conference (WWW)*, pages 307–318, Rio de Janeiro, Brazil, 2013.

Petros Dellaportas and Ioulia Papageorgiou. Multivariate mixtures of normals with unknown number of components. *Statistics and Computing*, 16(1):57–68, 2006.

Avinava Dubey, Ahmed Hefny, Sinead Williamson, and Eric P. Xing. A nonparametric mixture model for topic modeling over time. In *Proc. of the 13th SIAM International Conference on Data Mining (SDM)*, pages 530–538, 2013.

Adabriand Furtado, Nazareno Andrade, Nigini Oliveira, and Francisco Brasileiro. Contributor Profiles, their Dynamics, and their Importance in Five Q&A Sites. In *Proc. of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW)*, pages 1237–1252, 2013.

Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation. Using Multiple Sequences. *Statistical Science*, 7(4):457–511, 1992.

Samuel J. Gershman and David M. Blei. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(6):1–12, 2012.

Siddharth Gopal and Yiming Yang. Von Mises-Fisher Clustering Models. In *Proc. of the 31st International Conference on Machine Learning (ICML)*, pages 154–162, 2014.

John Hartigan and Manchek Wong. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied statistics*, 28(1):100–108, 1979.

W. Keith Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970.

Gregor Heinrich. Parameter Estimation for Text Analysis. Technical report version 2.9, vsonix GmbH + University of Leipzig, 2009.

Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. Unsupervised graph-based topic labelling using dbpedia. In *Proc. of the Sixth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 465–474, 2013.

John T. Kent. The Fisher-Bingham Distribution on the Sphere. *Journal of the Royal Statistical Society*, 44:71–80, 1982.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

Jan R. Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Economics*. John Wiley and Sons, NY, USA, 1988.

Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*. Academic Press Inc., London, UK, 2000.

Geoffrey McLachlan and David Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Agostino Nobile. Bayesian finite mixtures: a note on prior specification and posterior computation. Technical report, Department of Statistics, University of Glasgow, 2005.

Derek O'Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications (ESA)*, 42(13):5645–5657, 2015.

David Peel, William J. Whiten, and Geoffrey J. McLachlan. Fitting Mixtures of Kent Distributions to Aid in Joint Set Identification. *Journal of the American Statistical Association*, 96(453):56–63, 2001.

Xiangju Qin, Derek Greene, and Pádraig Cunningham. A latent space analysis of editor lifecycles in wikipedia. In *Proc. of the 5th International Workshop on Mining Ubiquitous and Social Environments (MUSE) at ECML/PKDD 2014*, pages 3–18, 2014.

Carl Edward Rasmussen. The Infinite Gaussian Mixture Model. *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 12:554–560, 2000.

Joseph Reisinger, Austin Waters, Bryan Silverthorn, and Raymond J. Mooney. Spherical Topic Models. In *Proc. of the 27th International Conference on Machine Learning (ICML)*, pages 903–910, 2010.

Sylvia Richardson and Peter J. Green. On Bayesian Analysis of Mixtures with an Unknown Number of Components. *Statistical Methodology*, 59(4):731–792, 1997.

Matthew Rowe. Mining User Lifecycles from Online Community Platforms and their Application to Churn Prediction. In *Proc. of the 13th IEEE International Conference on Data Mining (ICDM)*, pages 1–10, 2013.

Diane Mary Sculley. Web-Scale K-Means Clustering. In *Proc. of the 19th International World Wide Web Conference (WWW)*, pages 1177–1178, 2010.

Alexander Smola and Shravan Narayanamurthy. An architecture for parallel topic models. *Proc. of the VLDB Endowment*, 3(1-2):703–710, 2010.

Michael Spivak. *Calculus on Manifolds A Modem Approach to Classical Theorems of Advanced Calculus*. Addison-Wesley Publishing Company, USA, 1965.

Julian Straub, Trevor Campbell, Jonathan P. How, and John W. Fisher III. Small-Variance Nonparametric Clustering on the Hypersphere. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 334–342, 2015a.

Julian Straub, Jason Chang, Oren Freifeld, and John W. Fisher III. A Dirichlet Process Mixture Model for Spherical Data. In *Proc. of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 930–938, 2015b.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association (JASA)*, 101(476):1566–1581, 2006.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008.

Yi Wang, Shi-Xia Liu, Lizhu Zhou, and Hui Su. Mining Naturally Smooth Evolution of Clusters from Dynamic Data. In *Proc. of the Seventh SIAM International Conference on Data Mining (SDM)*, pages 125–134, 2007.

Greg C. G. Wei and Martin A. Tanner. A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, 81(411):699–704, 1990.

Chih-Ping Weia and I-Tang Chiub. Turning telecommunications call details to churn prediction: a data mining approach. *Expert Systems with Applications (ESA)*, 23(2):103–112, 2002.

Zhihua Zhang, Kap Luk Chan, Yiming Wu, and Chibiao Chen. Learning a multivariate Gaussian mixture model with the reversible jump MCMC algorithm. *Statistics and Computing*, 14(4): 343–355, 2004.

Mingjun Zhong and Mark Girolami. Reversible Jump MCMC for Non-Negative Matrix Factorization. In *Proc. of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 663–670, 2009.

Shi Zhong and Joydeep Ghosh. Generative Model-based Document Clustering: A Comparative. Study. *Knowledge and Information Systems (KAIS)*, 8(3):374–384, 2005.

Appendices

Appendix A Detailed Inference for Collapsed Gibbs Sampling

Updates for z_i. The conditional distribution for z_i is given by

$$
\begin{aligned}
\gamma(z_{ih}) \equiv P(z_{ih}=1|\mathcal{Z}_{-i},\{x_j\}_{j=1}^N,\kappa,m,\sigma^2,\mu_0,C_0,\alpha) \propto \\
&\int_\pi\int_\mu f(x_i|\mu_{z_i},\kappa_{z_i})P(z_i|\pi)f(\pi|\alpha)\prod_{j\neq i}P(z_j|\pi)f(x_j|\mu_{z_j},\kappa_{z_j})\prod_{h=1}^{H}f(\mu_h|\mu_0,C_0)\ f(\kappa_h|\ m,\sigma^2) \\
&\propto \int_\pi P(z_i|\pi)f(\pi|\alpha)\prod_{j\neq i}P(z_j|\pi)\times \\
&\qquad\prod_{h=1}^{H}\int_{\mu_h}\left(f(x_i|\mu_{z_i},\kappa_{z_i})\prod_{j\neq i,z_{jh}=1}f(x_j|\mu_h,\kappa_h)\right)f(\mu_h|\mu_0,C_0)\ f(\kappa_h|\ m,\sigma^2) \\
&\propto \int_\pi P(z_i|\pi)f(\pi|\alpha)\prod_{j\neq i}P(z_j|\pi)\times\prod_{h=1}^{H}\int_{\mu_h}\left(f(x_i|\mu_{z_i},\kappa_{z_i})\prod_{j\neq i,z_{jh}=1}f(x_j|\mu_h,\kappa_h)\right)f(\mu_h|\mu_0,C_0)
\end{aligned}
\tag{A-1}
$$

where the description of each term in Eq. (A-1) can be referred to Eq. (7). Expanding out the Dirichlet priors and the discrete distributions according to their usual definitions, i.e., $P(\pi|\alpha) \sim \text{Dirichlet}(H,\alpha)$, $P(z_i|\pi) \sim \text{Multi}(.|\pi)$, yields[18]:

$$
\begin{aligned}
\int_\pi P(z_{ih}=1|\pi)f(\pi|\alpha)\prod_{j\neq i}P(z_j|\pi) &= \int_\pi\prod_{h=1}^{H}\pi_h^{z_{ih}}\frac{\Gamma(H\alpha)}{\prod_{h=1}^{H}\Gamma(\alpha)}\prod_{h=1}^{H}\pi_h^{\alpha-1}\prod_{j\neq i}\prod_{h=1}^{H}\pi_h^{z_{ih}} \\
&= \frac{\Gamma(H\alpha)}{\prod_{h=1}^{H}\Gamma(\alpha)}\int_\pi\prod_{h=1}^{H}\pi_h^{\alpha+n_{h,-i}+z_{ih}-1} \\
&= \frac{\Gamma(H\alpha)}{\prod_{h=1}^{H}\Gamma(\alpha)}\frac{\prod_{h=1}^{H}\Gamma(\alpha+n_{h,-i}+z_{ih})}{\Gamma(H\alpha+N)} \\
&\propto \Gamma(\alpha+n_{h,-i}+1) \propto (\alpha+n_{h,-i})\Gamma(\alpha+n_{h,-i}) \propto (\alpha+n_{h,-i})
\end{aligned}
\tag{A-2}
$$

Similarly, expanding out the probabilities $f(x_i|\mu_{z_i},\kappa_{z_i})$, $f(x_j|\mu_h,\kappa_h)$ and $f(\mu_h|\mu_0,C_0)$ according to their usual definitions, $f(x_i|\mu_{z_i},\kappa_{z_i}) = C_D(\kappa_{z_i})\exp\{\kappa_{z_i}\mu_{z_i}^T x_i\}$ and $f(\mu_h|\mu_0,C_0) = C_D(C_0)\exp\{C_0\mu_0^T\mu_h\}$

18. In the calculation, following Heinrich (2009), the Dirichlet integral of the first kind for summation function, $\sum_h \pi_h$=1, $\frac{\Gamma(\alpha)^H}{\Gamma(H\alpha)}$= $\int_\pi \prod_{h=1}^{H}\pi_h^{\alpha_h-1}$ is used, analogous to the identity of the beta integral: $B(\alpha_1,\alpha_2)$= $\int_0^1 x^{\alpha_1-1}(1-x)^{\alpha_2-1}\,dx$. The identity $\Gamma(x+1)$=$x\Gamma(x)$ is used in the last line.

yields[19]:

$$\begin{aligned}
&\prod_{h=1}^{H}\int_{\mu_h}\left(f(x_i|\mu_{z_i},\kappa_{z_i})\prod_{j\neq i,z_{jh}=1}f(x_j|\mu_h,\kappa_h)\right)f(\mu_h|\mu_0,C_0) = \\
&\quad=\prod_{h=1}^{H}\int_{\mu_h}C_D(\kappa_h)^{n_{h,-i}+z_{ih}}C_D(C_0)\exp\left\{C_0\mu_0^T\mu_h+\kappa_h\mu_h^T\left(x_i+\sum_{j\neq i}z_{jh}x_j\right)\right\} \\
&\quad=\prod_{h=1}^{H}C_D(\kappa_h)^{n_{h,-i}+z_{ih}}C_D(C_0)\int_{\mu_h}\exp\left\{\left(C_0\mu_0^T+\kappa_h x_i^T+\kappa_h\sum_{j\neq i}z_{jh}x_j^T\right)\mu_h\right\} \\
&\quad=\prod_{h=1}^{H}\frac{C_D(\kappa_h)^{n_{h,-i}+z_{ih}}C_D(C_0)}{C_D(\|\kappa_h\left(x_i+\sum_{j\neq i}z_{jh}x_j\right)+C_0\mu_0\|)}
\end{aligned} \tag{A-3}$$

Substituting Eq. (A-2)-(A-3) in Eq. (A-1) yields the following:

$$\begin{aligned}
&P(z_{ih}=1|\mathcal{Z}_{-i},\{x_j\}_{j=1}^N,\kappa,m,\sigma^2,\mu_0,C_0,\alpha) \\
&\quad\propto\int_\pi P(z_i|\pi)f(\pi|\alpha)\prod_{j\neq i}P(z_j|\pi)\prod_{h=1}^{H}\int_{\mu_h}\left(f(x_i|\mu_{z_i},\kappa_{z_i})\prod_{j\neq i,z_{jh}=1}f(x_j|\mu_h,\kappa_h)\right)f(\mu_h|\mu_0,C_0) \\
&\quad\propto(\alpha+n_{h,-i})\prod_{h=1}^{H}\frac{C_D(\kappa_h)^{n_{h,-i}+z_{ih}}C_D(C_0)}{C_D(\|\kappa_h\left(x_i+\sum_{j\neq i}z_{jh}x_j\right)+C_0\mu_0\|)} \\
&\quad\propto(\alpha+n_{h,-i})C_D(\kappa_h)\frac{C_D(\|\kappa_h\sum_{j\neq i}z_{jh}x_j+C_0\mu_0\|)}{C_D(\|\kappa_h\left(x_i+\sum_{j\neq i}z_{jh}x_j\right)+C_0\mu_0\|)}
\end{aligned} \tag{A-4}$$

Updates for κ. Similarly, the conditional distribution for κ_h is given by

$$\begin{aligned}
f(\kappa_h|\mathcal{Z},\{x_j\}_{j=1}^N,\kappa,m,\sigma^2,\mu_0,C_0) &\propto\int_{\mu_h}\prod_{z_{ih}=1}f(x_i|\mu_h,\kappa_h)f(\mu_h|\mu_0,C_0)\ f(\kappa_h|\ m,\sigma^2) \\
&\propto\int_{\mu_h}\prod_{z_{ih}=1}C_D(\kappa_h)C_D(C_0)\exp\left\{\kappa_h\mu_h^Tx_i+C_0\mu_0^T\mu_h\right\}\text{logNormal}(\kappa_h|m,\sigma^2) \\
&\propto\int_{\mu_h}C_D(\kappa_h)^{n_h}C_D(C_0)\exp\left\{\kappa_h\mu_h^T\sum_{z_{ih}=1}x_i+C_0\mu_0^T\mu_h\right\}\text{logNormal}(\kappa_h|m,\sigma^2) \\
&\propto C_D(\kappa_h)^{n_h}\ C_D(C_0)\ \text{logNormal}(\kappa_h|m,\sigma^2)\ \int_{\mu_h}\exp\left\{\kappa_h\mu_h^T\sum_{z_{ih}=1}x_i+C_0\mu_0^T\mu_h\right\} \\
&\propto\frac{C_D(\kappa_h)^{n_h}C_D(C_0)}{C_D(\|\kappa_h\sum_{j:z_{jh}=1}z_{jh}x_j+C_0\mu_0\|)}\ \text{logNormal}(\kappa_h|m,\sigma^2)
\end{aligned} \tag{A-5}$$

where n_h is the number of observations assigned to the h-th component.

19. In this calculation, the identity of the von Mises-Fisher integral (Mardia and Jupp (2000), page 168; Chiuso and Picci (1998)) is used, $\int_{R^{D-1}}\exp\left\{\kappa\mu^Tx\right\}dx=(2\pi)^{\frac{D}{2}}\left(\frac{\hat{\kappa}}{2}\right)^{1-D/2}I_{D/2-1}(\hat{\kappa})=\frac{1}{C_D(\hat{\kappa})}$, where $\hat{\kappa}=\|\kappa\mu\|$.

Appendix B Empirical Bayes Estimates for Prior Parameters

The joint likelihood function of the prior parameters $(\mu_0, C_0, m, \sigma^2, \alpha)$ is given by:

$$\begin{aligned}\mathcal{L}(\mu_0, C_0, m, \sigma^2, \alpha | \{x_i\}_{i=1}^N, \mu, \kappa, \pi) &= f(\{x_i\}_{i=1}^N | \mu, \kappa, \pi) f(\mu | \mu_0, C_0) f(\kappa | m, \sigma^2) f(\pi | \alpha) \\ &\propto \prod_{h=1}^H C_D(C_0) \exp\left\{C_0 \mu_0^T \mu_h\right\} \prod_{h=1}^H \frac{1}{\kappa_h \sigma \sqrt{2\pi}} \exp\left\{-\frac{(\log \kappa_h - m)^2}{2\sigma^2}\right\} \frac{\Gamma(H\alpha)}{\prod_{h=1}^H \Gamma(\alpha)} \prod_{h=1}^H \pi_h^{\alpha-1}\end{aligned} \tag{B-1}$$

Since the prior parameters are assumed to be independent, we have the following log-likelihood functions according to Eq. (B-1)

$$\begin{aligned}\log \mathcal{L}(C_0, \mu_0 | \{x_i\}_{i=1}^N, \mu) &= H \log C_D(C_0) + C_0 \mu_0^T \left(\sum_{h=1}^H \mu_h\right) \\ \log \mathcal{L}(m, \sigma^2 | \{x_i\}_{i=1}^N, \kappa) &= -\frac{H}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{h=1}^H \left(\log(\kappa_h)^2 - 2m \log(\kappa_h) + m^2\right) \\ \log \mathcal{L}(\alpha | \{x_i\}_{i=1}^N, \pi) &= -\log\left(\frac{\Gamma(H\alpha)}{\prod_{h=1}^H \Gamma(\alpha)}\right) + (\alpha - 1) \prod_{h=1}^H \pi_h\end{aligned} \tag{B-2}$$

Following Gopal and Yang (2014), the empirical Bayes estimate for C_0, μ_0 is given by:

$$\arg\max_{\mu_0, C_0} \; H \log C_D(C_0) + C_0 \mu_0^T \left(\sum_{h=1}^H \mu_h\right) \tag{B-3}$$

which suggests the following updates (Gopal and Yang, 2014)

$$\mu_0 = \frac{\sum_{h=1}^H \mu_h}{\| \sum_{h=1}^H \mu_h \|}, \; C_0 = \frac{\bar{r} D - \bar{r}^3}{1 - \bar{r}^2}, \text{ where}, \bar{r} = \frac{\| \sum_{h=1}^H \mu_h \|}{H} \tag{B-4}$$

Similarly, the empirical Bayes estimate for α is given by Gopal and Yang (2014):

$$\arg\max_{\alpha > 0} \; -\log\left(\frac{\Gamma(H\alpha)}{\prod_{h=1}^H \Gamma(\alpha)}\right) + (\alpha - 1) \sum_{h=1}^H \pi_h \tag{B-5}$$

Since there exists no closed-form solution for Eq. (B-5), we rely on numerical optimization such as gradient descent to find the Maximum Likelihood Estimate for α.

The empirical Bayes estimate for m, σ^2 can be obtained in a similar way by taking the partial derivative on Eq. (B-2) w. r. t. m and σ^2, respectively, which gives:

$$\begin{aligned}\frac{\partial \log \mathcal{L}(m, \sigma^2 | \mathcal{X}, \kappa)}{\partial m} &= -\frac{1}{2\sigma^2} \sum_{h=1}^H \left[-2\log(\kappa_h) + 2m\right] = 0 \\ \frac{\partial \log \mathcal{L}(m, \sigma^2 | \mathcal{X}, \kappa)}{\partial x} &= -\frac{H}{2x} + \frac{1}{2x^2} \sum_{h=1}^H \left(\log(\kappa_h)^2 - 2m \log(\kappa_h) + m^2\right) = 0 \, (\text{Let x}{=}\sigma^2) \\ \Rightarrow m = \frac{1}{H} \sum_{h=1}^H \log(\kappa_h), \; x &= \sigma^2 = \frac{1}{H} \sum_{h=1}^H \log(\kappa_h)^2 - m^2\end{aligned} \tag{B-6}$$

Alternatively, we can use the Monte Carlo Expectation-Maximum (MCEM) algorithm (Wei and Tanner, 1990) to estimate the prior parameters m and σ^2.

- E-step: Randomly sample L times the value for κ_h, $\{\kappa_h^{(i)}\}_{h=1}^H, i \in [1, L]$. Note that previously, we use MCMC sampling to estimate κ_ks. Here, we can reuse the set of generated κ_hs in the MCMC step.
- M-step: Estimate m and σ^2 by maximising the log-likelihood:

$$m = \frac{1}{LH}\sum_{i=1}^{L}\sum_{h=1}^{H}\log(\kappa_h^{(i)}),\ \sigma^2 = \frac{1}{LH}\sum_{i=1}^{L}\sum_{h=1}^{H}\log(\kappa_h^{(i)})^2 - m^2 \tag{B-7}$$

Appendix C Factorization of Acceptance Probability

We can factorize $P(H, \mathcal{Z}, \{\pi_h, \theta_h\}_{h=1}^H, \mathcal{B}|\{x_i\}_{i=1}^N)$ below:

$$\begin{aligned} P(H, \mathcal{Z}, \{\pi_h, \theta_h\}_{h=1}^H, \mathcal{B}|\{x_i\}_{i=1}^N) &= \frac{P(H, \mathcal{Z}, \pi, \mu, \kappa, m, \sigma^2, \mu_0, C_0, \alpha, \{x_i\}_{i=1}^N)}{P(\{x_i\}_{i=1}^N)} \\ &= \frac{P(H)f(\pi|\alpha)P(\mathcal{Z}|\pi)f(\mu|C_0, \mu_0)f(\kappa|m, \sigma^2)f(\{x_i\}_{i=1}^N|\mathcal{Z}, \mu, \kappa)}{P(\{x_i\}_{i=1}^N)} \end{aligned} \tag{C-1}$$

where each term in Eq. (C-2) can be specified as follows:

$$\begin{aligned} \frac{P(H+1, \mathcal{Z}', \Theta', \mathcal{B}|\{x_i\}_{i=1}^N)}{P(H, \mathcal{Z}, \Theta, \mathcal{B}|\{x_i\}_{i=1}^N)} = \frac{P(H+1)}{P(H)} \times \frac{f(\pi'|\alpha)}{f(\pi|\alpha)} \times \frac{P(\mathcal{Z}'|\pi')}{P(\mathcal{Z}|\pi)} \times \frac{f(\mu'|\mu_0, C_0)}{f(\mu|\mu_0, C_0)} \times \\ \frac{f(\kappa'|m, \sigma^2)}{f(\kappa|m, \sigma^2)} \times \frac{f(\{x_i\}_{i=1}^N|\mathcal{Z}', \mu', \kappa')}{f(\{x_i\}_{i=1}^N|\mathcal{Z}, \mu, \kappa)} \end{aligned} \tag{C-2}$$

$$\begin{aligned} \frac{P(H+1)}{P(H)} &= \frac{f(H+1;1)}{f(H;1)}, \quad \frac{P(\{z_i'\}_{i=1}^N|\pi)}{P(\{z_i\}_{i=1}^N|\pi)} = \frac{\pi_{j_1}^{n_{j_1}}\pi_{j_2}^{n_{j_2}}}{\pi_{j_*}^{n_{j_*}}} \\ \frac{f(\pi'|\alpha)}{f(\pi|\alpha)} &= \frac{\frac{\Gamma((H+1)\alpha)}{(\Gamma(\alpha))^{H+1}}\prod_{h=1}^{H+1}\pi_h^{\alpha-1}}{\frac{\Gamma(H\alpha)}{(\Gamma(\alpha))^H}\prod_{h=1}^{H}\pi_h^{\alpha-1}} = \frac{1}{B(\alpha, H\alpha)}\frac{\pi_{j_1}^{\alpha-1}\pi_{j_2}^{\alpha-1}}{\pi_{j_*}^{\alpha-1}} \\ \frac{f(\mu'|C_0, \mu_0)}{f(\mu|C_0, \mu_0)} &= (H+1)\frac{f(\mu_{j_1}|C_0, \mu_0)f(\mu_{j_2}|C_0, \mu_0)}{f(\mu_{j_*}|C_0, \mu_0)} \\ \frac{f(\kappa'|m, \sigma^2)}{f(\kappa|m, \sigma^2)} &= \frac{f(\kappa_{j_1}|m, \sigma^2)f(\kappa_{j_2}|m, \sigma^2)}{f(\kappa_{j_*}|m, \sigma^2)} \\ \frac{f(\{x_i\}_{i=1}^N|\{z_i'\}_{i=1}^N, \mu', \kappa')}{f(\{x_i\}_{i=1}^N|\{z_i\}_{i=1}^N, \mu, \kappa)} &= \text{(likelihood ratio)} = \frac{\prod_{i=1}^N f(x_i|\mu_{z_i'}', \kappa_{z_i'}')}{\prod_{i=1}^N f(x_i|\mu_{z_i}, \kappa_{z_i})} \\ \text{where } n_{j_1} = \sum_{i=1}^N z_{ij_1}, n_{j_2} &= \sum_{i=1}^N z_{ij_2}, n_{j_*} = n_{j_1} + n_{j_1} \end{aligned} \tag{C-3}$$

where $B(\cdot, \cdot)$ is the Beta function, the $(H+1)$-factor in the third line being the ratio $(H+1)!/H!$ from the order statistics densities for the parameters (π, μ, κ) (i.e. label switching for the parameters). The calculation of the Jacobian matrix J and its determinant is similar to that in Zhang et al. (2004) and given in Appendix D.

Following Dellaportas and Papageorgiou (2006), the Jacobian term $\left|\frac{\partial\Sigma}{\partial(\lambda,V)}\right|$ can be computed by using the following formulae

$$\partial\lambda = V_d^{'}(\partial\Sigma)V_d,\ \partial V = (\lambda_d I_D - \Sigma)^{+}(\partial\Sigma)V_d \tag{C-4}$$

where λ_d and V_d, d=1, $\cdots$, D, are the specific eigenvalue-eigenvector pairs of Σ; $(A)^{+}$ denotes the Moore-Penrose pesudo-inverse matrix of A (See Magnus and Neudecker (1988) p.179). For symmetric perturbations, Magnus and Neudecker (1988) (p.181) suggested that applying the properties of vec operator (i.e. vec ABC = $(C^{'} \otimes A)$ vec B) and the chain rule, Eq. (C-4) can be rewritten as follows

$$\partial\lambda = (V_d^{'} \otimes V_d^{'})\mathbf{D}\partial v(\Sigma),\ \partial V = (V_d^{'} \otimes (\lambda_d I_D - \Sigma)^{+})\mathbf{D}\partial v(\Sigma) \tag{C-5}$$

where $\mathbf{D}$ is the duplication matrix (see Magnus and Neudecker (1988) Chapter 3). From Eq. (C-5), we obtain the derivatives

$$\begin{aligned}\frac{\partial\lambda}{\partial(\text{vec }\Sigma)^{'}} &= V_d^{'} \otimes V_d^{'} \\ \frac{\partial V}{\partial(\text{vec }\Sigma)^{'}} &= V_d^{'} \otimes (\lambda_d I_D - \Sigma)^{+}\end{aligned} \tag{C-6}$$

According to the inverse function theorem (Spivak, 1965), the inverse of the Jacobian matrix of an invertible function is equivalent to the Jacobian matrix of the inverse function. Specifically, if the Jacobian of the function $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and non-singular at the point $p \in \mathbb{R}^n$, then F is invertible in some neighbourhood of p and we have

$$J_{F^{-1}}(F(p)) = (J_F(p))^{-1} \tag{C-7}$$

Thus, if $\left|\frac{\partial(\lambda,V)}{\partial(\text{vec }\Sigma)^{'}}\right|$ is non-zero, then we can have

$$\left|\frac{\partial\Sigma}{\partial(\lambda,V)}\right| = \left|\frac{\partial(\lambda,V)}{\partial(\text{vec }\Sigma)^{'}}\right|^{-1} \tag{C-8}$$

Appendix D Calculation of Jacobian Matrix

Let $s = \{\pi_{j_*}, \mu_{j_*}, g_{j_*}\}$ and $s' = \{\pi_{j_1}, \pi_{j_2}, \mu_{j_1}, \mu_{j_2}, g_{j_1}, g_{j_2}\}$ denote the state of Markov chain before and after the split move, respectively, where $g_{j_*} = (\lambda_{j_*1}, \cdots, \lambda_{j_*D})^T$, $g_{j_1} = (\lambda_{j_1 1}, \cdots, \lambda_{j_1 D})^T$, $g_{j_2} = (\lambda_{j_2 1}, \cdots, \lambda_{j_2 D})^T$. Denote the set of continuous random variables needed for the split move as $u = \{u_1, u_2, u_3\}$, where $u_2 = (u_{21}, \cdots, u_{2D})^T$, $u_3 = (u_{31}, \cdots, u_{3D})^T$. Thus, from the transformation defined by Eq. (17), we can obtain the Jacobian matrix J for the split move (from (s, u) to s') as follows (Zhang et al., 2004)

$$J = \frac{\partial s'}{\partial(s,u)} = \begin{bmatrix} \frac{\partial\pi_{j_1}}{\partial\pi_{j_*}} & \frac{\partial\pi_{j_1}}{\partial u_1} & \frac{\partial\pi_{j_1}}{\partial\mu_{j_*}} & \frac{\partial\pi_{j_1}}{\partial u_2} & \frac{\partial\pi_{j_1}}{\partial g_{j_*}} & \frac{\partial\pi_{j_1}}{\partial u_3} \\ \frac{\partial\pi_{j_2}}{\partial\pi_{j_*}} & \frac{\partial\pi_{j_2}}{\partial u_1} & \frac{\partial\pi_{j_2}}{\partial\mu_{j_*}} & \frac{\partial\pi_{j_2}}{\partial u_2} & \frac{\partial\pi_{j_2}}{\partial g_{j_*}} & \frac{\partial\pi_{j_2}}{\partial u_3} \\ \frac{\partial\mu_{j_1}}{\partial\pi_{j_*}} & \frac{\partial\mu_{j_1}}{\partial u_1} & \frac{\partial\mu_{j_1}}{\partial\mu_{j_*}} & \frac{\partial\mu_{j_1}}{\partial u_2} & \frac{\partial\mu_{j_1}}{\partial g_{j_*}} & \frac{\partial\mu_{j_1}}{\partial u_3} \\ \frac{\partial\mu_{j_2}}{\partial\pi_{j_*}} & \frac{\partial\mu_{j_2}}{\partial u_1} & \frac{\partial\mu_{j_2}}{\partial\mu_{j_*}} & \frac{\partial\mu_{j_2}}{\partial u_2} & \frac{\partial\mu_{j_2}}{\partial g_{j_*}} & \frac{\partial\mu_{j_2}}{\partial u_3} \\ \frac{\partial g_{j_1}}{\partial\pi_{j_*}} & \frac{\partial g_{j_1}}{\partial u_1} & \frac{\partial g_{j_1}}{\partial\mu_{j_*}} & \frac{\partial g_{j_1}}{\partial u_2} & \frac{\partial g_{j_1}}{\partial g_{j_*}} & \frac{\partial g_{j_1}}{\partial u_3} \\ \frac{\partial g_{j_2}}{\partial\pi_{j_*}} & \frac{\partial g_{j_2}}{\partial u_1} & \frac{\partial g_{j_2}}{\partial\mu_{j_*}} & \frac{\partial g_{j_2}}{\partial u_2} & \frac{\partial g_{j_2}}{\partial g_{j_*}} & \frac{\partial g_{j_2}}{\partial u_3} \end{bmatrix} \tag{D-1}$$

From the transformation in Eq. (17), we calculate the partial derivatives:

$$
\begin{array}{llllllllllll}
\frac{\partial \pi_{j_1}}{\partial \pi_{j_*}} & = & u_1, & \frac{\partial \pi_{j_1}}{\partial u_1} & = & \pi_{j_*}, & \frac{\partial \pi_{j_1}}{\partial \mu_{j_*}} & = & 0_{1\times D}, & \frac{\partial \pi_{j_1}}{\partial u_2} & = & 0_{1\times D} \\
\frac{\partial \pi_{j_1}}{\partial g_{j_*}} & = & 0_{1\times D}, & \frac{\partial \pi_{j_1}}{\partial u_3} & = & 0_{1\times D} & & & & & & \\
\frac{\partial \pi_{j_2}}{\partial \pi_{j_*}} & = & 1-u_1, & \frac{\partial \pi_{j_2}}{\partial u_1} & = & -\pi_{j_*}, & \frac{\partial \pi_{j_2}}{\partial \mu_{j_*}} & = & 0_{1\times D}, & \frac{\partial \pi_{j_2}}{\partial u_2} & = & 0_{1\times D} \\
\frac{\partial \pi_{j_2}}{\partial g_{j_*}} & = & 0_{1\times D}, & \frac{\partial \pi_{j_2}}{\partial u_3} & = & 0_{1\times D} & & & & & & \\
\frac{\partial \mu_{j_1}}{\partial \pi_{j_*}} & = & 0_{D\times 1}, & \frac{\partial \mu_{j_1}}{\partial u_1} & = & 0_{D\times 1}, & \frac{\partial \mu_{j_1}}{\partial \mu_{j_*}} & = & I, & \frac{\partial \mu_{j_1}}{\partial u_3} & = & 0_{D\times D} \\
\frac{\partial \mu_{j_2}}{\partial \pi_{j_*}} & = & 0_{D\times 1}, & \frac{\partial \mu_{j_2}}{\partial u_1} & = & 0_{D\times 1}, & \frac{\partial \mu_{j_2}}{\partial \mu_{j_*}} & = & I, & \frac{\partial \mu_{j_2}}{\partial u_3} & = & 0_{D\times D} \\
\frac{\partial g_{j_1}}{\partial \pi_{j_*}} & = & 0_{D\times 1}, & \frac{\partial g_{j_1}}{\partial u_1} & = & 0_{D\times 1}, & \frac{\partial g_{j_1}}{\partial \mu_{j_*}} & = & 0_{D\times D} & & & \\
\frac{\partial g_{j_2}}{\partial \pi_{j_*}} & = & 0_{D\times 1}, & \frac{\partial g_{j_2}}{\partial u_1} & = & 0_{D\times 1}, & \frac{\partial g_{j_2}}{\partial \mu_{j_*}} & = & 0_{D\times D} & & &
\end{array}
\tag{D-2}
$$

The other partial derivatives can be calculated as:

$$
\begin{aligned}
&\frac{\partial \mu_{j_1}}{\partial u_{2d}} = -\sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}}\lambda_{j_*d}^{\frac{1}{2}} V_{j_*d}, && \frac{\partial \mu_{j_1}}{\partial \lambda_{j_*d}} = -\frac{1}{2}\sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}}\lambda_{j_*d}^{-\frac{1}{2}} u_{2d} V_{j_*d} \\
&\frac{\partial \mu_{j_2}}{\partial u_{2d}} = \sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}}\lambda_{j_*d}^{\frac{1}{2}} V_{j_*d}, && \frac{\partial \mu_{j_2}}{\partial \lambda_{j_*d}} = \frac{1}{2}\sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}}\lambda_{j_*d}^{-\frac{1}{2}} u_{2d} V_{j_*d} \\
&\frac{\partial \lambda_{j_1 d}}{\partial u_{2l}} = \begin{cases} -2u_{3d}u_{2d}\lambda_{j_*d}\frac{\pi_{j_*}}{\pi_{j_1}} & l = d, \\ 0 & l \neq d \end{cases} \\
&\frac{\partial \lambda_{j_2 d}}{\partial u_{2l}} = \begin{cases} -2(1-u_{3d})u_{2d}\lambda_{j_*d}\frac{\pi_{j_*}}{\pi_{j_2}} & l = d, \\ 0 & l \neq d \end{cases} \\
&\frac{\partial \lambda_{j_1 d}}{\partial \lambda_{j_* l}} = \begin{cases} u_{3d}(1-u_{2d}^2)\frac{\pi_{j_*}}{\pi_{j_1}} & l = d, \\ 0 & l \neq d \end{cases} \\
&\frac{\partial \lambda_{j_2 d}}{\partial \lambda_{j_* l}} = \begin{cases} (1-u_{3d})(1-u_{2d}^2)\frac{\pi_{j_*}}{\pi_{j_2}} & l = d, \\ 0 & l \neq d \end{cases} \\
&\frac{\partial \lambda_{j_1 d}}{\partial u_{3l}} = \begin{cases} \lambda_{j_*d}(1-u_{2d}^2)\frac{\pi_{j_*}}{\pi_{j_1}} & l = d, \\ 0 & l \neq d \end{cases} \\
&\frac{\partial \lambda_{j_2 d}}{\partial u_{3l}} = \begin{cases} -\lambda_{j_*d}(1-u_{2d}^2)\frac{\pi_{j_*}}{\pi_{j_2}} & l = d, \\ 0 & l \neq d, \quad d \in [1, D] \end{cases}
\end{aligned}
\tag{D-3}
$$

Therefore, we have the following expressions

$$
\begin{array}{llllllllllll}
\frac{\partial \mu_{j_1}}{\partial u_2} & = & -\sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}} V_{j_*} \Lambda_{j_*}^{\frac{1}{2}}, & \frac{\partial \mu_{j_1}}{\partial g_{j_*}} & = & -\frac{1}{2}\sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}} V_{j_*} \Lambda_{j_*}^{-\frac{1}{2}} U_2 & & & & & & \\
\frac{\partial \mu_{j_2}}{\partial u_2} & = & \sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}} V_{j_*} \Lambda_{j_*}^{\frac{1}{2}}, & \frac{\partial \mu_{j_2}}{\partial g_{j_*}} & = & \frac{1}{2}\sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}} V_{j_*} \Lambda_{j_*}^{-\frac{1}{2}} U_2 & & & & & & \\
\frac{\partial g_{j_1}}{\partial u_2} & = & -2\frac{\pi_{j_*}}{\pi_{j_1}} \Lambda_{j_*} U_3 U_2, & \frac{\partial g_{j_1}}{\partial g_{j_*}} & = & \frac{\pi_{j_*}}{\pi_{j_1}} U_3 (1-U_2^2), & \frac{\partial g_{j_1}}{\partial u_3} & = & \frac{\pi_{j_*}}{\pi_{j_1}} \Lambda_{j_*} (I-U_2^2) \\
\frac{\partial g_{j_2}}{\partial u_2} & = & 2\frac{\pi_{j_*}}{\pi_{j_2}} \Lambda_{j_*} (U_3 - I) U_2, & \frac{\partial g_{j_2}}{\partial g_{j_*}} & = & \frac{\pi_{j_*}}{\pi_{j_2}} (I-U_3)(I-U_2^2), & \frac{\partial g_{j_2}}{\partial u_3} & = & \frac{\pi_{j_*}}{\pi_{j_2}} \Lambda_{j_*} (U_2^2 - I)
\end{array}
\tag{D-4}
$$

where $U_2 = diag(u_{2d}, \cdots, u_{2D})$ and $U_3 = diag(u_{31}, \cdots, u_{3D})$ are diagonal matrices. Substituting Eq. (D-2) and (D-4) into Eq. (D-1) yields the Jacobian matrix, J, as follows:

$$J = \begin{bmatrix} u_1 & \pi_{j_*} & \mathbf{0}_{1\times D} & \mathbf{0}_{1\times D} & \mathbf{0}_{1\times D} & \mathbf{0}_{1\times D} \\ 1-u_1 & -\pi_{j_*} & \mathbf{0}_{1\times D} & \mathbf{0}_{1\times D} & \mathbf{0}_{1\times D} & \mathbf{0}_{1\times D} \\ \mathbf{0}_{D\times 1} & \mathbf{0}_{D\times 1} & \mathbf{I} & -\sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}} V_{j_*} \Lambda_{j_*}^{\frac{1}{2}} & -\frac{1}{2}\sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}} V_{j_*} \Lambda_{j_*}^{-\frac{1}{2}} U_2 & \mathbf{0}_{D\times D} \\ \mathbf{0}_{D\times 1} & \mathbf{0}_{D\times 1} & \mathbf{I} & \sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}} V_{j_*} \Lambda_{j_*}^{\frac{1}{2}} & \frac{1}{2}\sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}} V_{j_*} \Lambda_{j_*}^{-\frac{1}{2}} U_2 & \mathbf{0}_{D\times D} \\ \mathbf{0}_{D\times 1} & \mathbf{0}_{D\times 1} & \mathbf{0}_{D\times D} & -2\frac{\pi_{j_*}}{\pi_{j_1}} \Lambda_{j_*} U_3 U_2 & \frac{\pi_{j_*}}{\pi_{j_1}} U_3 (I - U_2^2) & \frac{\pi_{j_*}}{\pi_{j_1}} \Lambda_{j_*} (I - U_2^2) \\ \mathbf{0}_{D\times 1} & \mathbf{0}_{D\times 1} & \mathbf{0}_{D\times D} & 2\frac{\pi_{j_*}}{\pi_{j_2}} \Lambda_{j_*} (U_3 - I) U_2 & \frac{\pi_{j_*}}{\pi_{j_2}} (I - U_3)(I - U_2^2) & \frac{\pi_{j_*}}{\pi_{j_2}} \Lambda_{j_*} (U_2^2 - I) \end{bmatrix} \tag{D-5}$$

where $\mathbf{0}_{D\times 1}$ is the $D \times 1$ zero vector, $\mathbf{0}_{1\times D}$ is the $1 \times D$ zero vector, $\mathbf{0}_{D\times D}$ the $D \times D$ zero matrix, U_2=diag$\{u_{21}, u_{22}, \cdots, u_{2D}\}$ and U_3=diag$\{u_{31}, u_{32}, \cdots, u_{3D}\}$ are diagonal matrices.

By blocking the Jacobian matrix J defined by Eq. (D-5), we have

$$|det(J)| = \pi_{j_*} \cdot |det(J_1)| \tag{D-6}$$

where

$$J_1 = \left[\begin{array}{cc|cc} \mathbf{I} & -\sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}} V_{j_*} \Lambda_{j_*}^{\frac{1}{2}} & -\frac{1}{2}\sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}} V_{j_*} \Lambda_{j_*}^{-\frac{1}{2}} U_2 & \mathbf{0}_{D\times D} \\ \mathbf{I} & \sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}} V_{j_*} \Lambda_{j_*}^{\frac{1}{2}} & \frac{1}{2}\sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}} V_{j_*} \Lambda_{j_*}^{-\frac{1}{2}} U_2 & \mathbf{0}_{D\times D} \\ \hline \mathbf{0}_{D\times D} & -2\frac{\pi_{j_*}}{\pi_{j_1}} \Lambda_{j_*} U_3 U_2 & \frac{\pi_{j_*}}{\pi_{j_1}} U_3 (I - U_2^2) & \frac{\pi_{j_*}}{\pi_{j_1}} \Lambda_{j_*} (I - U_2^2) \\ \mathbf{0}_{D\times D} & 2\frac{\pi_{j_*}}{\pi_{j_2}} \Lambda_{j_*} (U_3 - I) U_2 & \frac{\pi_{j_*}}{\pi_{j_2}} (I - U_3)(I - U_2^2) & \frac{\pi_{j_*}}{\pi_{j_2}} \Lambda_{j_*} (U_2^2 - I) \end{array}\right] \tag{D-7}$$

We partitioned J_1 into $J_1 = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$ as indicated by the vertical and horizontal lines in Eq. (D-7). When J_{11} is invertible, according to Theorem by Brualdi and Schneider (1983), we have

$$|det(J_1)| = |det(J_{11})| \cdot |det(J_{22} - J_{21} J_{11}^{-1} J_{12})| \tag{D-8}$$

Calculating each determinant in Eq. (D-8) and substituting them into Eq. (D-6) yields the absolute of determinant of J, $det(J)$, as follows

$$|det(\mathbf{J})| = \frac{\pi_{j_*}^{3D+1}}{(\pi_{j_1}\pi_{j_2})^{\frac{3D}{2}}} \prod_{d=1}^{D} \lambda_{j_* d}^{3/2} (1 - u_{2d}^2) \tag{D-9}$$

Appendix E Comparison of Different Split-merge Moves

In this appendix, we compared the performance of TvMF mixture model with common eigenvectors (i.e. simplified RJMCMC move) and without common eigenvectors (i.e. original RJMCMC move) for split-merge moves on synthetic data from three aspects: (1) clustering performance, (2) acceptance rate for the moves and posterior estimation for the number of components H, and (3) the trace plot of the log likelihood and number of components over sweeps (in Figure 12). Table 6 compares the clustering performance and posterior estimation of H for TvMFMM with two different strategies for split-merge moves.

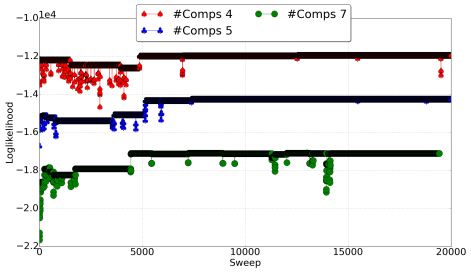
Table 6: Clustering performance and posterior estimation of H for TvMFMM with/without common eigenvectors for split-merge moves on synthetic data.

True H	ARI	AMI	NMI	Homogeneity	Completeness
TvMFMM with common eigenvectors for split-merge move					
4	0.925	0.898	0.899	0.898	0.899
5	0.947	0.941	0.945	0.947	0.944
7	0.809	0.821	0.829	0.827	0.832
TvMFMM without common eigenvectors for split-merge move					
4	0.931	0.904	0.904	0.904	0.904
5	0.936	0.931	0.935	0.936	0.934
7	0.792	0.812	0.82	0.814	0.826

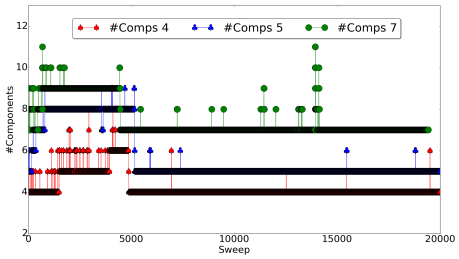
True H	Portion of moves accepted (%)		Posterior estimation for H
	Split-merge	Birth-death	
TvMFMM with common eigenvectors for split-merge move			
4	0.89	0.15	P(4)=0.997, P(5)=0.002, P(6)=0.001
5	0.25	0.16	P(5)=0.999, P(6)=0.001
7	0.69	0.3	P(7)=0.977, P(8)=0.020, P(9)=0.002, P(10)=0.001
TvMFMM without common eigenvectors for split-merge move			
4	0.08	0.05	P(4)=1.000
5	0.02	0.1	P(5)=0.999, P(6)=0.001
7	0.42	0.11	P(7)=0.855, P(8)=0.143, P(9)=0.001

We observe from Table 6 that TvMFMM with and without common eigenvectors for split-merge moves present very similar clustering performance on the three datasets; TvMFMM with common eigenvectors for split-merge move achieves relatively better clustering performance. Both versions of TvMFMM give an accurate estimation of the number of components with the highest posterior probability for the true value of H. In addition, TvMFMM with both settings show very low acceptance rate for the split-merge and birth-death moves. This is one of the characteristics of the RJMCMC algorithm, which generally has very low acceptance rate for the moves, as observed by other researchers (Richardson and Green, 1997; Zhang et al., 2004; Dellaportas and Papageorgiou, 2006).

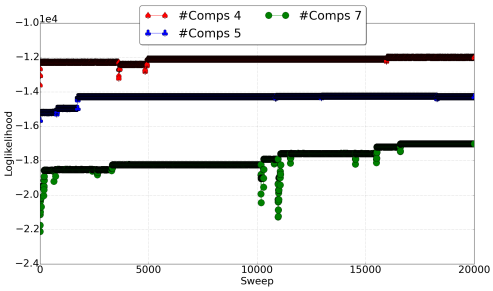
Figure 12 shows that TvMFMM with/without common eigenvectors for split-merge moves present similar mixing properties for the RJMCMC chains in terms of the smoothness of the log likelihood and number of components over sweeps. It is obvious that the RJMCMC chain becomes stable gradually in terms of the number of components and log likelihood after 5000 sweeps. We actually choose to use a burn-in period of 10000 sweeps for comparison of clustering performance on synthetic data. Overall, the results suggest that TvMFMM with/without common eigenvectors for the split-merge moves show similar performance and mixing rates on synthetic data. Therefore, for the results presented in the empirical evaluation, we ran TvMFMM and OTvMFMM with common eigenvectors for split-merge moves.



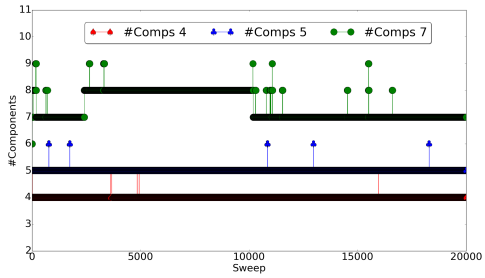
(a) Log likelihood for simplified RJMCMC move



(b) #Components for simplified RJMCMC move



(c) Log likelihood for the original RJMCMC move



(d) #Components for the original RJMCMC move

Figure 12: Trace plot of the log likelihood and estimated number of components for TvMFMM with simplified RJMCMC move and with original RJMCMC move on synthetic data.

Appendix F Model Parameter Analysis

To choose the optimal number of clusters for parametric models, we used NMF model with different number of clusters to analyse Wikipedia quarterly datasets, and measured cluster coherence using normalised pairwise mutual information (NPMI). The results are presented in Figure 13. Recall that the NPMI metric captures the semantic interpretability of discovered clusters based on the corresponding descriptor terms. Higher coherence scores indicate better semantic interpretability, thus more coherent and interpretable topics. One obvious trend in Figure 13 is that NMF models running with larger number of clusters result in lower values of NPMI scores, indicating less coherent and interpretable clusters; whereas models running with smaller number of clusters have higher values of NPMI scores, suggesting more coherent and interpretable clusters. Moreover, NMF models running with 5, 10 and 15 clusters generate very close NPMI scores on all the quarterly datasets, of which models with 10 clusters output the best overall NPMI scores. Therefore, we choose $H = 10$ for parametric models.

We perform further analysis of the log likelihood and cluster coherence of Bayesian vMF mixture model on selected Wikipedia quarterly datasets for varying number of clusters, which are given in Figure 14. As the number of clusters increase from 5 to 20, the log likelihood of the model on the training and held-out datasets also increases, after which the log likelihood becomes relatively stable for any increase in #clusters. On the other hand, as #clusters increases from 5 to 10, the NPMI scores also go up, after which the NPMI scores reduce slightly and become relatively stable. This suggests that larger numbers of clusters does not indicate improved cluster coherence

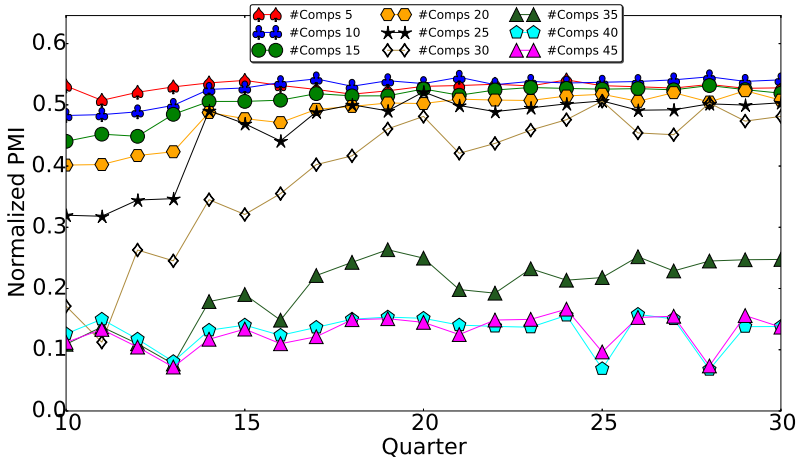
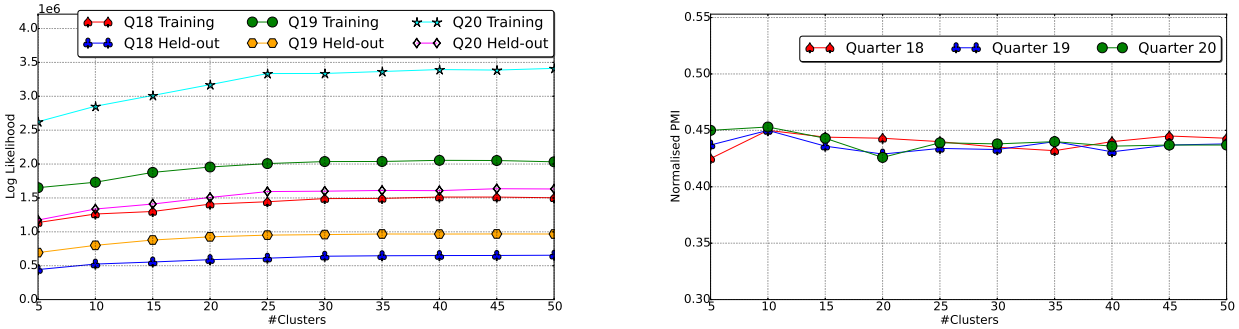


Figure 13: Cluster coherence (NPMI) of NMF model for varied number of clusters ($k \in [5, 45]$) on Wikipedia quarterly training datasets.

/ interpretability. The analysis provides further support for our choice of $H = 10$ for parametric models.



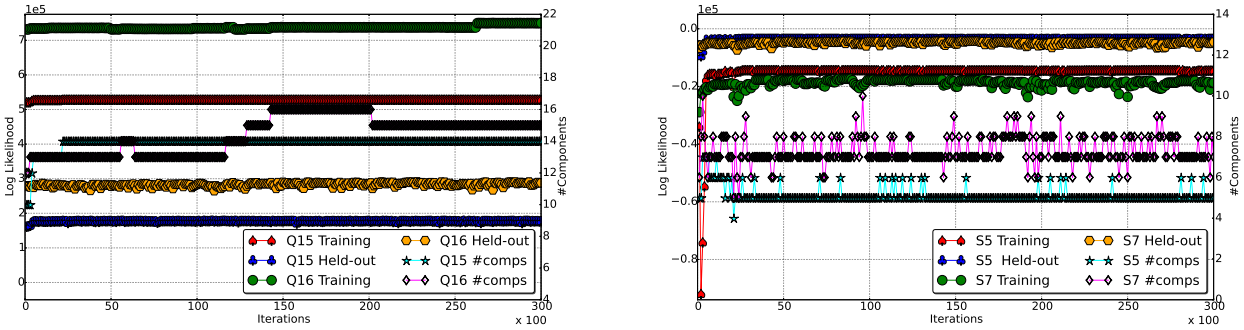
(a) Log likelihood

(b) Topic coherence NPMI

Figure 14: Log likelihood and cluster coherence for Bayesian vMFMM with varied number of clusters ($k \in [5, 50]$) on the 18th (Q18), 19th (Q19) and 20th (Q20) quarter of Wikipedia editor dataset.

One main advantage of the reversible jump MCMC algorithm is the ability to explore multiple models simultaneously, which brings side-benefit that can refine the inappropriate initialization of model parameters. To explore this point, we present a plot of log likelihood and estimated number of components of TvMFMM on selected Wikipedia and synthetic datasets after every 100 iterations. The results are presented in Figure 15, from which we observe that: the log likelihood of the model increases in the beginning, but then becomes relatively stable as more iterations proceed; there are some fluctuations in the log likelihood corresponding to synthetic training dataset with 7

components. The number of components increases with the iterations in the beginning, and then experiences some fluctuations with more iterations. By checking the output of model statistics at these points, we notice that the fluctuation points correspond to the accepted split-merge / birth-death moves where the model explores alternative models. In addition, the statistics of TvMFMM on Wikipedia data and synthetic data with 5 components shows better mixing property (i.e. less fluctuations) than those on synthetic data with 7 components. The results are consistent with our statement about the advantage of RJMCMC.



(a) Wikipedia data

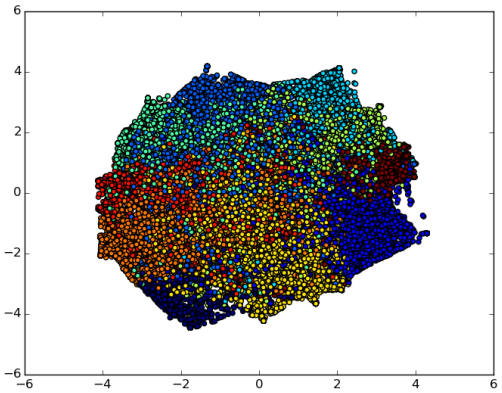
(b) Synthetic data

Figure 15: Trace plot of log likelihood and estimated number of components for TvMFMM after each 100 iterations on the 15th (Q15) and 16th (Q16) quarter of Wikipedia editor datasets, and synthetic datasets with 5 (S5) and 7 (S7) components. Where the left yaxis corresponds to log likelihood and the right yaxis represents #components.

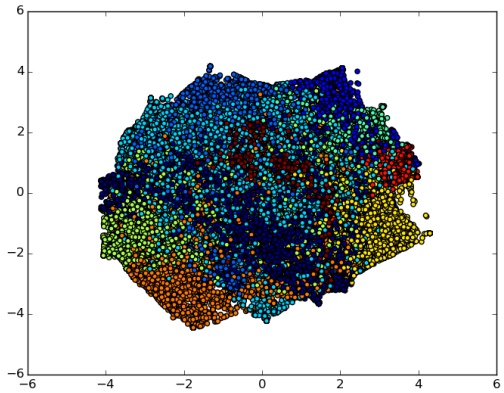
Appendix G Discriminative Analysis for Different Models

Embedding methods, such as t-distributed stochastic neighbourhood embedding (t-SNE; van der Maaten and Hinton (2008)) can be used to visualise high-dimensional data in a two or three-dimensional map. This visualisation provides a unique insight into the discriminative ability of mixture models in terms of separating data points in low-dimensional representation.

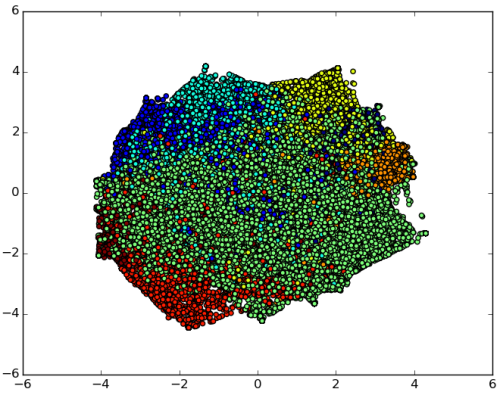
Figure 16 presents a 2D embedding of the inferred topic estimation by five models, using the t-SNE method, where each dot represents an entry of user behavioural data and each color-shape represents a topic. Visually, the proposed OTvMFMM produces a relatively better separation of data points than NMF, DTM and BvMFMM, while DP-GMM does not produce a well-separated embedding, and data points assigned to different clusters tend to mix together. This is consistent with our qualitative analysis in Section 5.2.2 that OTvMFMM can produce more interpretable and intuitive topics than other models. Intuitively, a well-separated representation is more discriminative for data separation.



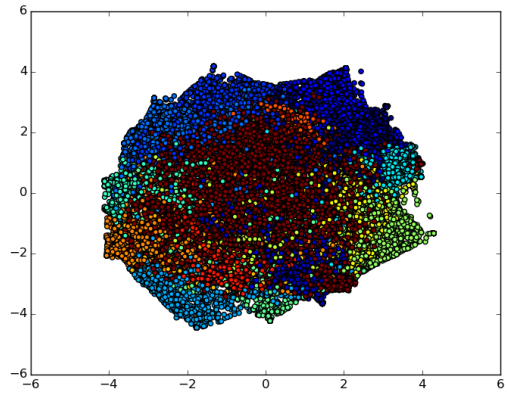
(a) NMF (Quarter 18)



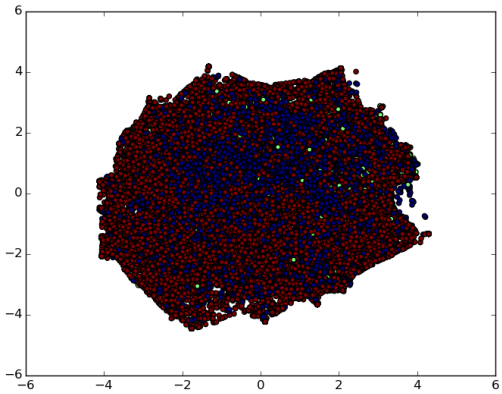
(b) DTM (Quarter 18)



(c) BvMFMM (Quarter 18)



(d) OTvMFMM (Quarter 18)

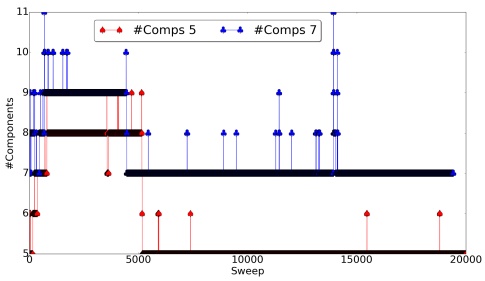


(e) DP-GMM (Quarter 18)

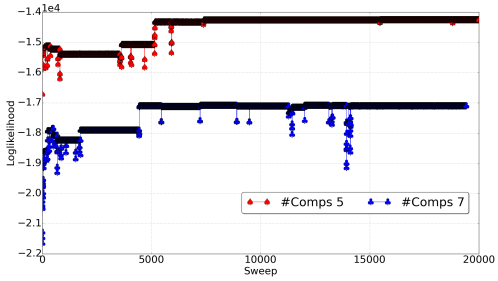
Figure 16: t-SNE 2D embedding of the topical representations by different models on the 18th quarter of Wikipedia Editor dataset.

Appendix H Effects of Prior Parameters

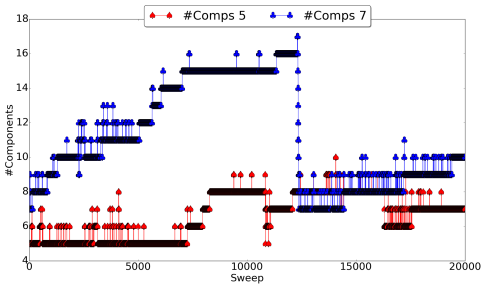
The prior parameters m and σ^2 control the range of the concentration parameters κ, where the value of κ affects the mixing property / convergence of the RJMCMC chain. Figure 17 presents the trace plot of the number of components and log likelihood for TvMFMM with different values for m and σ^2 on synthetic data with 5 and 7 components. The values of m and σ^2 are chosen so that the corresponding ranges of κ are able to illustrate the effects of appropriate and inappropriate priors on the convergence of the chain. If the priors m and σ^2 are appropriately set, the convergence of RJMCMC algorithm can be sped up, leading to well mixing chain, as observed in in Figure 17 (a-b) that the chains begin converging from around the 5000th sweep onwards. On the other hand, when the priors m and σ^2 are inappropriately set, the convergence of RJMCMC algorithm can be slowed down, resulting in poor mixing chain, as obvious in Figure 17 (c-d) that the chains experience more fluctuations in the number of components and log likelihood over sweeps. The observations suggest that the trace plot can be used to diagnose whether the priors are appropriately set.



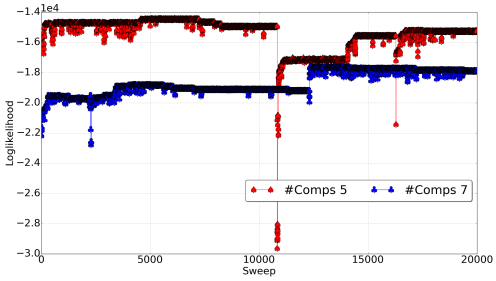
(a) #Components - appropriate prior: $m = 21.0$, $\sigma^2 = 1.025$, CI for κ is (19.4, 22.8)



(b) Log likelihood - appropriate prior: $m = 21.0$, $\sigma^2 = 1.025$, CI for κ is (19.4, 22.8)



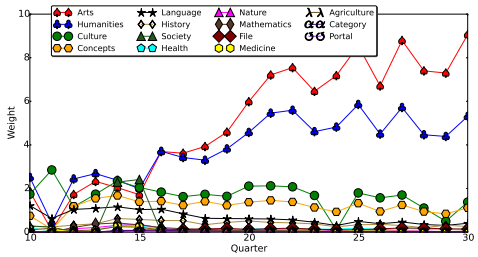
(c) #Components - inappropriate prior: $m = 28.0$, $\sigma^2 = 1.035$, CI for κ is (25.0, 31.4)



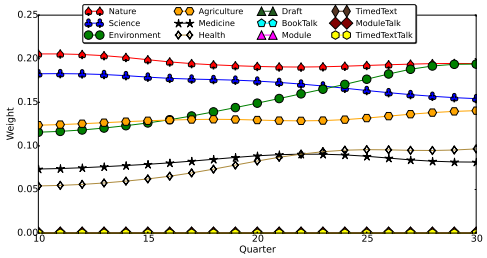
(d) Log likelihood - inappropriate prior: $m = 28.0$, $\sigma^2 = 1.035$, CI for κ is (25.0, 31.4)

Figure 17: Trace plot of log likelihood and the number of components over iterations for TvMFMM with appropriate/inappropriate values for the prior parameters m and σ^2 on synthetic data, including 99.9% confidence interval (CI) for κ.

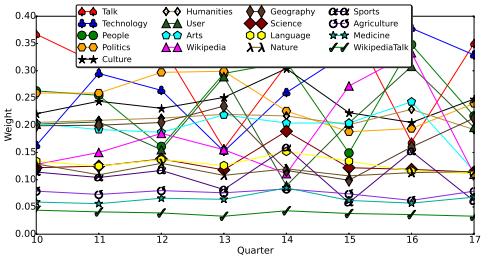
Appendix I Dynamics of Top Terms for User Roles – *Content Editors*



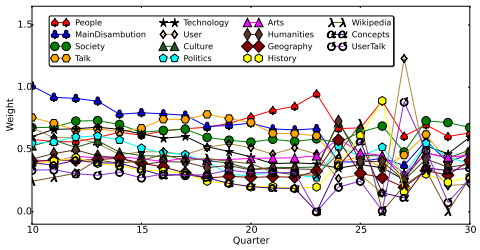
(a) NMF (Misc Content Editors I)



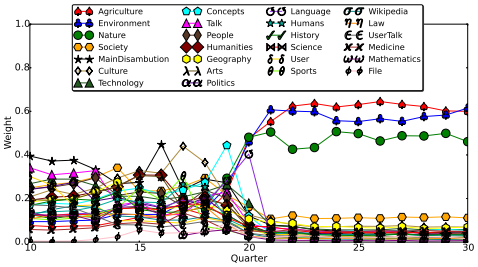
(b) DTM (All-round contributors II)



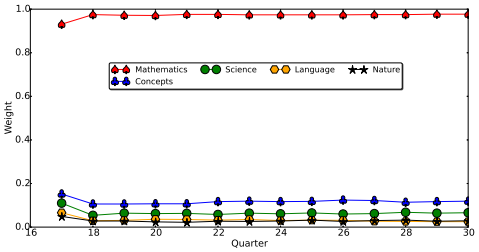
(c) BvMFMM (All-round contributors)



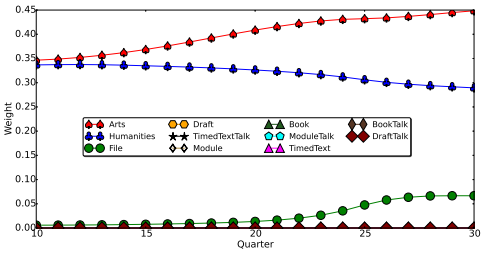
(d) DP-GMM (All-round contributors II)



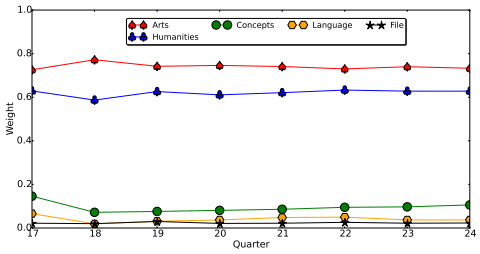
(e) OTvMFMM (All-round contributors)



(f) OTvMFMM (Misc Mathematics)



(g) DTM (Misc Arts Humanities)



(h) OTvMFMM (Arts Humanities)

Figure 18: Evolution of top terms for common user roles (*Content Editors*) identified by different models.