

A Bayesian Mixture Model with Linear Regression Mixing Proportions

Xiuyao Song¹, Chris Jermaine¹, Sanjay Ranka¹, John Gums²

¹Department of Computer and Information Sciences and Engineering

²Departments of Pharmacy Practice and Family Medicine
University of Florida

Gainesville, FL, USA, 32611

xsong,cjermain,ranka@cise.ufl.edu, jgums@ufl.edu

ABSTRACT

Classic mixture models assume that the prevalence of the various mixture components is fixed and does not vary over time. This presents problems for applications where the goal is to learn how complex data distributions evolve. We develop models and Bayesian learning algorithms for inferring the temporal trends of the components in a mixture model as a function of time. We show the utility of our models by applying them to the real-life problem of tracking changes in the rates of antibiotic resistance in *Escherichia coli* and *Staphylococcus aureus*. The results show that our methods can derive meaningful temporal antibiotic resistance patterns.

Categories and Subject Descriptors

G.3 [Mathematics of Computing]: PROBABILITY AND STATISTICS—*reliability*; H.2.8 [DATABASE MANAGEMENT]: Database Applications—*algorithms*

General Terms

Reliability, Algorithms

1. INTRODUCTION

Motivation. This paper addresses the problem of clustering time series data and using regression modeling techniques to learn the change in prominence of each cluster over time. Our motivating application is as follows. A hospital laboratory collects specimens of *E. coli* bacteria from patients, then cultures the bacteria and tests the antibiotic resistance patterns of each *E. coli* specimen. For a particular antibiotic drug, a lower dosage threshold and an upper dosage threshold are first defined by the pharmacist. If a dosage less than the lower threshold successfully kills the bacteria, then the *E. coli* isolate is *susceptible* to the drug. If the dosage required to kill the bacteria is larger than the upper threshold, then the *E. coli* isolate is *resistant* to the drug. There is a third possibility that the bacteria is killed when the dosage falls between the lower threshold and upper threshold. In such a situation, we say the resistance is *undetermined*. A typical resulting data set is given in Table 1.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.
Copyright 2008 ACM 978-1-60558-193-4/08/08 ...\$5.00.

Table 1: Antibiotic resistance data of *E. coli* on multiple drugs (R: Resistant; S: Susceptible; U: Undetermined).

Patient ID	Cefuroxime	Ampicillin	Amp/Sulbactam	Amox/K Clay	Nitrofurantoin	Imipenem	Collect Date
1	S	S	S	S	S	S	9/26/2004
2	S	R	U	S	S	S	12/1/2004
3	R	R	R	U	S	S	1/12/2006
...							

Each specimen (or patient) is represented as a row vector. The first column is the patient ID, the last column is the date of specimen collection; all of the middle columns are the test results.

Given such a data set, an epidemiologist often wants to perform two key tasks. First, it is useful to determine whether subsets of the *E. coli* specimens exhibit substantially different multiple antibiotic resistance patterns, because it classifies the specimens into different subsets or strains whose resistance profiles—and hence the threat to public health that they represent—are similar. Second, it is useful to determine how these subsets change over time. Usually, the antibiotic resistance is not static because bacteria have mutated to circumvent the effects of antibiotics due to selective pressures. Tougher bacteria strains survive and reproduce, and it is useful to know whether more resistant strains are increasing in prevalence.

There are many ways to perform the segmentation task. For example, it is useful to use a mixture model to segment the data into several clusters. Each cluster represents a unique antibiotic resistance pattern, either exhibited by a single *E. coli* strain or shared by multiple bacteria strains. However, the second task is more difficult. In mixture-model-based clustering, a mixing proportion is associated with each cluster, which reflects the prevalence of the corresponding resistance pattern. The mixing proportions of the clusters are fixed with respect to time since the time information is ignored when the mixture model is trained.

A Linear Regression Mixture Model. In this paper, we present a Bayesian mixture model which takes temporal information into account to perform trend analysis when clustering data.

Reconsider the data set given in Table 1. Assume the number of antibiotic drugs is D and the data set is generated by a mixture model with K clusters. The mixing proportion is denoted by $\vec{\pi} = \langle \pi_1, \pi_2, \dots, \pi_K \rangle$. Because an epidemiologist is interested in tracking the prevalence of all the bacterial strains over time, the generative model for each cluster remains unchanged over time, but the mixing proportion changes according to $\vec{\pi}(t)$. In our method, we assume that $\vec{\pi}$ models linear trends. Mathematically, the mixing

proportion at any time t is expressed as:

$$\bar{\pi}(t) = \bar{\pi}(t_b) + (t - t_b) \times \frac{\bar{\pi}(t_e) - \bar{\pi}(t_b)}{t_e - t_b}, \quad (1)$$

where $\bar{\pi}(t_b)$ and $\bar{\pi}(t_e)$ is the mixing proportion at the beginning and the end of time, respectively. The term $\frac{\bar{\pi}(t_e) - \bar{\pi}(t_b)}{t_e - t_b}$ gives the slope vector for the mixing proportion. Thus, when we track the bacterial strains with a linear regression mixture model, we will obtain not only a clustering, but also a set of trends.

We use a linear model for two reasons. First, it is simple and informative. Linear regression can effectively smooth noise and eliminate the short-term oscillations in the resistance pattern so that epidemiologists can immediately identify those bacteria strains that rise (or fall) in prevalence over time.

Second, linear regression has the benefit of consistency. The mixing proportions over all clusters should sum to 1. If $\sum \bar{\pi}(t_b) = 1$ and $\sum \bar{\pi}(t_e) = 1$ are satisfied at the beginning and the end of time, Eqn. (1) will guarantee that $\sum \bar{\pi}(t) = 1$ at any time t . Without linear regression, we have to perform normalization to make sure the mixing proportions sum to 1. In this case, even if each of the numerators are a smooth and intuitive curve, the mixing proportion after normalization will likely become erratic and hard to interpret.

Our Contributions. We propose a Bayesian mixture model with linear regression mixing proportions and provide two instantiations of our model: one with continuous Gaussian components, the other with discrete multinomial components. We use a Gibbs Sampler to learn the joint distribution of the random variables in the mixture model. We derive the conditional posterior distributions of the variables, which is technically difficult due to the regression model. Experiments show the utility of our models in tracking changes in the rates of antibiotic resistance in *Escherichia coli* and *Staphylococcus aureus*. The results show that our methods can derive meaningful antibiotic resistance patterns and their temporal trends.

Paper Organization. In Section 2, we describe the generative statistical model. In Section 3, the Markov Chain Monte Carlo method is used to approximate the parameters in the Bayesian mixture model. In Section 4, we give two instances of the proposed Bayesian mixture model. In Section 5, we experimentally test the mixture model. Section 6 discusses the related work and Section 7 concludes the paper.

2. STATISTICAL GENERATIVE MODEL

2.1 Generative Model

This section describes the statistical model we use for modeling the linear regression trend of the mixing components. Our discussion assumes that some parametric distribution f_z has been chosen as the generative model for the cluster indexed by the integer z . As described in the introduction, the mixing proportion is a linear function of the time t , so the probability or likelihood that we observe a data point y at time t is given by

$$f(y|t) = \sum_z \bar{\pi}_z(t) \times f_z(y).$$

Since $f(y|t)$ is conditioned on t , our generative model assumes that to generate a data point, a time stamp t is taken as input, and used along with the model parameter set Θ to generate the data point. The elements of Θ are summarized in Table 2. Given the linear regression, \vec{b} and \vec{e} can be used to determine the mixing proportion at any given time t by simple interpolation.

Table 2: Model parameters Θ

Symbol	Description
z	index of component in the mixture model
f_z	the parametric distribution of data specific to component z
$\bar{\pi}(t)$	the multinomial distribution of mixing proportions at time t
\vec{b}	the multinomial distribution of mixing proportions at the beginning of time
\vec{e}	the multinomial distribution of mixing proportions at the end of time
η_b	the parameter of Dirichlet prior of \vec{b}
η_e	the parameter of Dirichlet prior of \vec{e}

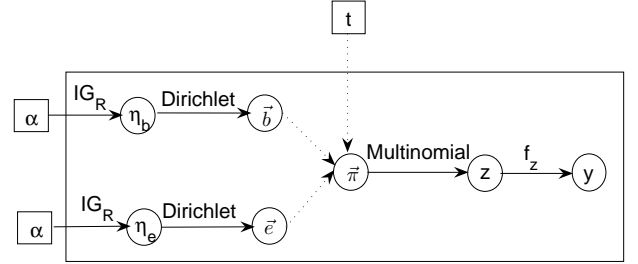


Figure 1: The generative model. A circle denotes a random variable. A rectangle denotes an input to the model. An arrow denotes a sampling process from an associated distribution. A dotted arrow denotes a calculation.

The generative model is given in Figure 1. The corresponding generating process of a data point is as follows:

1. Obtain hyperparameters of the mixing proportions:
 - (a) Draw η_b from inverse-Gamma with parameter α^1 ;
 - (b) Draw η_e from inverse-Gamma with parameter α ;
2. Obtain the end-point mixing proportions:
 - (a) Draw \vec{b} from a Dirichlet with parameter η_b ;
 - (b) Draw \vec{e} from a Dirichlet with parameter η_e ;
3. Draw a component z from multinomial $\bar{\pi}(t)$, where
$$\bar{\pi}(t) = \vec{b} + \frac{t - t_b}{t_e - t_b} \times (\vec{e} - \vec{b}).$$
4. Draw a data point y from f_z .

In general, f_z can be any parametric distribution. The parameterization of the regression mixture model is then:

$$\begin{aligned} \eta_b | \alpha &\sim IG_R(\alpha) \\ \eta_e | \alpha &\sim IG_R(\alpha) \\ \vec{b} | \eta_b &\sim \text{Dirichlet}(\eta_b) \\ \vec{e} | \eta_e &\sim \text{Dirichlet}(\eta_e) \\ z | \vec{b}, \vec{e}, t &\sim \text{multinomial}\left(\vec{b} + \frac{t - t_b}{t_e - t_b} \times (\vec{e} - \vec{b})\right) \\ y | f_z &\sim f_z \end{aligned}$$

¹Refer to the inverse-Gamma distribution defined in [19]. For simplicity, we use a constant vector $\alpha = \langle 1, 1 \rangle$ in experiments.

2.2 Comparison with Distribution Over Time

The linear regression mixture model takes the form of $f(y|t)$, and the time information is used as the input to the generative process. This is not the only way to make use of temporal information. One obvious alternative is given in [21], where the time is treated as a random variable. That is, the generative process is responsible for generating both the data point and the associated time. The PDF of such an alternative model takes the form of $f(y \wedge t)$.

Why do we prefer $f(y|t)$ over $f(y \wedge t)$ when assessing the prevalence trend of mixing components over time? The reason is that the model $f(y \wedge t)$ must be used to learn the distribution of the data as well as the timestamp distribution, so it is very sensitive to the distribution of time. If there are different amounts of data for different time periods—generally there is less data for earlier time periods—a very different model will be learned than if the amount of data is constant over time. Since in our application, an expert is interested in the relationship between time and the prevalence of each cluster, it is unacceptable to have the learned model depend fundamentally on how the data are distributed over time. A PDF of the form $f(y|t)$ removes the dependence on the time distribution because it is *conditioned* on the input time.

The differences between these alternative models can be clearly demonstrated by the following simple experiment. Assume a regression mixture model with three uni-dimensional Gaussian clusters, denoted by $C_1 = \mathcal{N}(-1, 0.2)$, $C_2 = \mathcal{N}(0, 0.2)$ and $C_3 = \mathcal{N}(1, 0.2)$. The mixing proportion evolves over time as follows:

$$\begin{aligned}\pi_1(t) &= 0.8 \times (t - t_b)/(t_e - t_b); \\ \pi_2(t) &= 0.2; \\ \pi_3(t) &= 0.8 - \pi_1(t).\end{aligned}\quad (2)$$

To generate a time series data set from such a regression mixture model, we repeat the following four steps:

- Step 1:* Draw a time sample t from the time distribution and calculate the mixing proportion at time t ;
- Step 2:* Randomly choose a cluster C_i according to these mixing proportions;
- Step 3:* Randomly draw a sample y from Gaussian C_i ;
- Step 4:* Produce a data point $\langle y, t \rangle$ and put it into the data set.

We generate two data sets by the above process. In the first data set, time t has an uniform distribution on $[t_b, t_e]$. In the second data set, t has a skewed Beta distribution over $[t_b, t_e]$. We use an EM algorithm [9] to learn a Gaussian mixture model for $f(y \wedge t)$ on these two data sets. Figure 2 gives the plots of the trained models.

The first model presented in Figure 2(a) does a relatively poor job of capturing the three Gaussian clusters. If the learned centroids are projected on “data y ” dimension, we obtain -0.6 , -0.3 and 0.3 , corresponding to the real centroids of $-1, 0$ and 1 respectively, so there is quite a bit of error. However, at least the clusters are ordered correctly along the timeline. According to functions given in Equation (2), the “real” cluster 1 starts with weight 0 at the beginning of time and shifts to 0.8 at the end of the time; thus, it should be centered later on the timeline. On the other hand, the “real” cluster 3 starts at weight 0.8 and goes to 0, so it should be centered earlier in the timeline. In Figure 2(a), the learned clusters are presented in the expected partial order along the timeline.

After shifting the time distribution towards the high side of the timeline using a Beta distribution, the learned model presented in Figure 2(b) gets even worse. All three cluster centroids shift toward the high side of the timeline. The model ignores the early stage

data because there are fewer data points in the early time. Furthermore, the learned centroids along the “data” dimension shift to -0.8 , -0.6 and -0.3 respectively. All of them are dragged downward because more samples are taken at the end of the timeline; the “real” cluster 1, which has the lowest value, is the most prevalent cluster at the end of the timeline. Therefore, compared to the data set in Figure 2(a), there are more data points generated by the “real” cluster 1.

The above experiment illustrates clearly how sensitive a model for $f(y \wedge t)$ can be to the distribution of the data’s timestamps. For the purpose of comparison, we also train a model based upon $f(y|t)$ over the two data sets. The model was learned using the methods that will be described in detail later in this paper. Figure 3 gives the results when time is uniformly distributed. Figure 4 gives the result when time has a Beta distribution. Comparing these two training results, we see that even if the training data sets have completely different time distributions, the Gaussian clusters and the trends for the mixing proportions captured by $f(y|t)$ are very similar. In other words, the learned $f(y|t)$ model is independent of the time distribution in time series data sets.

Finally, we mention that theoretically, we can transform the $f(y \wedge t)$ model to our model by applying Bayes’ rule: $f(y|t) = \frac{f(y \wedge t)}{f(t)}$. This would remove the dependence on the distribution of time. However, this approach has two drawbacks. First, it requires us to provide the distribution $f(t)$. Second, since the training algorithm for $f(y \wedge t)$ is tailored to the specific joint model, even if we apply Bayes’ rule to transform the joint model to a conditional model, the resulting model will not be as accurate as if we apply our conditional model directly. Referring to Figure 5, it is obvious that the transformed conditional model does a poor job of depicting the PDF of the “real” data set. Because of these two drawbacks of applying Bayes’ rule to remove the time dependence, our conditional model is a better choice.

3. MCMC METHODS

For the parametric mixture model proposed in Section 2.1, a popular estimation technique used is Maximum Likelihood Estimation (MLE). MLE provides the point estimate for the parameters that maximize the probability of the generated data. From a statistical point of view, the MLE method is considered to be robust (with some exceptions) and yields estimators with good statistical properties. Unfortunately, the MLE corresponding to our model is computationally intractable using classical methods such as EM. This is caused by the fact that the mixing proportion is not a single random variable. Rather, it is a non-linear function of two independent random variables (\vec{b} and \vec{e}), which makes deviation of suitable a EM algorithm almost impossible because the M-step requires a difficult, non-linear optimization.

As an alternative, we can use Bayesian methods to compute the posterior distributions of the random variables. First, we specify the *prior* distribution for each parameter in the mixture model. We then apply Bayes’ rule to convert the expression of likelihood of the parameters into the posterior probability distribution of the parameters. The conversion is given in Eqn. (3):

$$p(\Theta|\mathbf{y}) = \frac{p(\mathbf{y}|\Theta)p(\Theta)}{p(\mathbf{y})} \propto L(\Theta|\mathbf{y})p(\Theta), \quad (3)$$

where $p(\Theta)$ is the prior distribution of the parameter. $L(\Theta|\mathbf{y})$ is the likelihood of the parameter based on the observed data.

Unfortunately, transforming the product term in the right of Eqn. (3) into the posterior distribution in the form of a probability density function is often computationally difficult due to the multi-

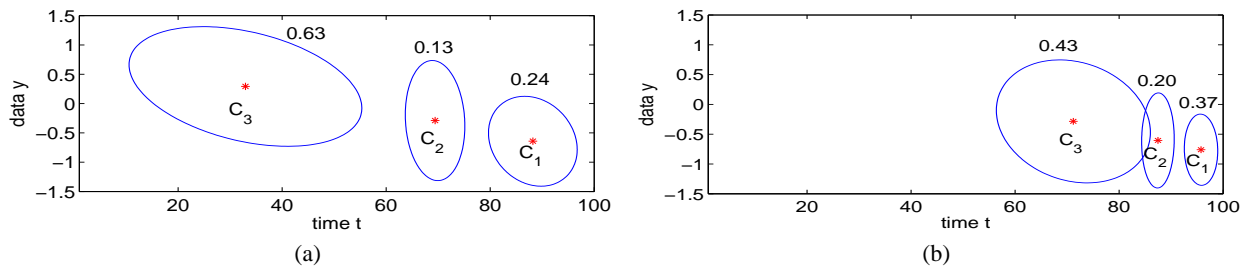


Figure 2: $f(y \wedge t)$ model trained over two data sets. (a) Time has a uniform distribution. (b) Time has a beta distribution. Both results are 3-component GMMs. The star denotes the centroid. The ellipse illustrates the covariance matrix. The fractional number above each cluster denotes its mixing proportion.

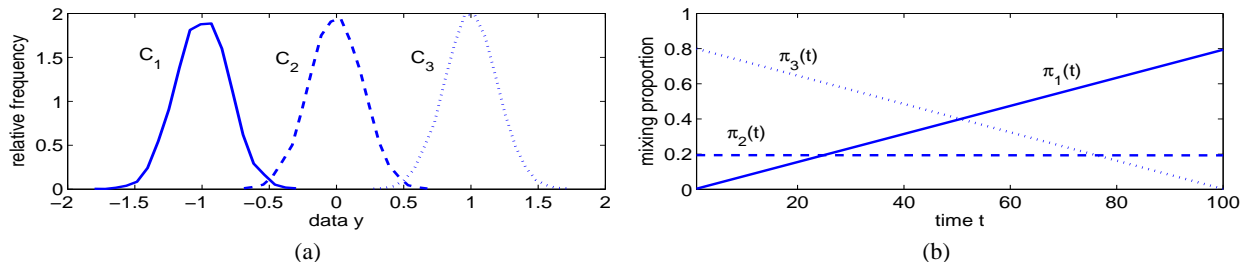


Figure 3: $f(y|t)$ model trained over data set with uniform time distribution. (a) denotes the PDF curves of the learned Gaussian clusters. (b) denotes the learned weight trend over time.

dimensional integrals. To work around this difficulty, *Markov Chain Monte Carlo* (MCMC) methods are widely used. MCMC methods are the techniques for sampling from the probability distribution by constructing a Markov chain whose stationary distribution is the distribution of interest. By repeatedly simulating the state of the chain, the method simulates samples drawn from the distribution of interest. In our case, each step of the Markov chain consists of all the parameters in the mixture model so the state of the chain is actually the joint distribution of all the model parameters.

The Gibbs sampler [13] is perhaps the most popular MCMC method. The key idea behind the Gibbs sampler is that at each step of the Markov chain, we only need to consider a univariate conditional distribution instead of a joint multivariate distribution. That means, at each step, we simulate the conditional distribution of one parameter assuming that all the other parameters are assigned fixed values. Such conditional distributions are far easier to simulate than complex joint distributions, and usually have simple forms. Assume there are p parameters. These p parameters are simulated from the p univariate conditional distributions sequentially rather than from a single p -variate joint distribution in a single pass. Suppose the parameter set Θ can be written as $\Theta = (\Theta_1, \dots, \Theta_p)$, where each Θ_i could be either unidimensional or multidimensional. Assume the univariate conditional density for Θ_i is $f_i(\theta_i | \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_p)$, $i = 1, 2, \dots, p$. In the Gibbs sampler, given all the parameters at step t , $\theta^{(t)} = (\theta_1^{(t)}, \dots, \theta_p^{(t)})$, the i parameters at step $t + 1$ are generated sequentially by i steps:

- (1) $\Theta_1^{(t+1)} \sim f_1(\theta_1 | \theta_2^{(t)}, \dots, \theta_p^{(t)})$,
- (2) $\Theta_2^{(t+1)} \sim f_2(\theta_2 | \theta_1^{(t+1)}, \theta_3^{(t)}, \dots, \theta_p^{(t)})$,
- ...
- (i) $\Theta_i^{(t+1)} \sim f_p(\theta_i | \theta_1^{(t+1)}, \dots, \theta_{i-1}^{(t+1)})$.

Each parameter is updated in turn from its posterior conditional distribution given all other parameters. The entire distribution for

all the parameters is explored as the number of steps of Gibbs sampling grows large.

4. TWO INSTANTIATIONS

In this section, we define two Bayesian mixture models, one with multinomial components to model discrete categorical data, the other with Gaussian components to model the continuous data. We also present derivations of two Gibbs Samplers, one suitable for learning each model.

4.1 Multinomial Model

As in the application to antibiotic resistance patterns given in Section 1, sometimes it is reasonable to assume that the data point is generated from a mixture model with multinomial components. Assume the dimensions are independent of each other. Let D be the number of dimensions and M be the number of category values on each dimension. Thus, a data point y_i is a D -ary vector, and each element takes the value from 1 to M . The multinomial mixture model with K components on n observations $\mathbf{y} = \{y_1, \dots, y_n\}$ can be written as:

$$p(\mathbf{y}_i | \vec{\pi}, \vec{\mu}) = \sum_{k=1}^K \pi_k \left[\prod_{j=1}^D \prod_{h=1}^M (\mu_k^{jh})^{x_i^{jh}} \right], \quad (4)$$

where $\vec{\pi} = \{\pi_1, \dots, \pi_k\}$ are the mixing proportions of components and $\vec{\mu} = \{\mu_1, \dots, \mu_k\}$ are the centroids of K components. The centroid of each component is a $D \times M$ matrix. Specifically, μ_k^{jh} denotes the element on the j^{th} row and h^{th} column in matrix μ_k , which is the probability of observing value h on j^{th} dimension in cluster k . $x_i^{jh} = 1$ if y_i has value h on j^{th} dimension, otherwise $x_i^{jh} = 0$. Note that we assume that K is user-given; there has been much research which can be directly applied addressing the problem of choosing the number of mixing components [20], though for brevity, we do not consider the issue of automatically choosing the number of mixture components in this paper.

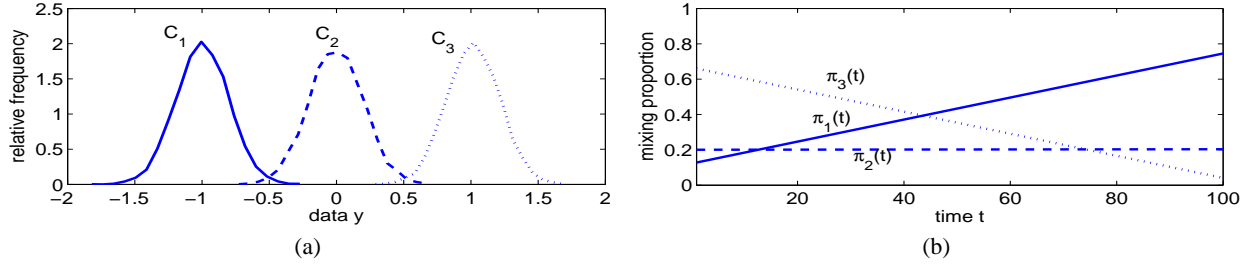


Figure 4: $f(y|t)$ model trained over data set where time has a beta distribution. (a) denotes the PDF curves of the learned Gaussian clusters. (b) denotes the learned weight trend over time.

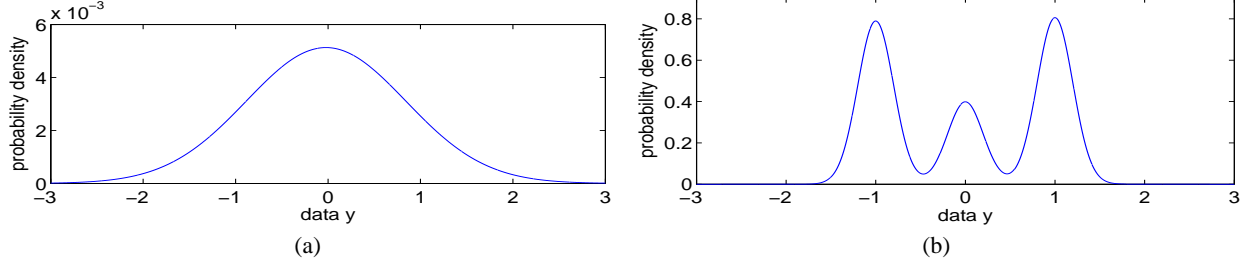


Figure 5: Applying Bayes' rule to remove the dependence of joint model on time. (a) is the PDF of the transformed conditional model at $t = 50$. Time has a Beta distribution in the corresponding joint model. (b) is the PDF of the "real" GMM model at $t = 50$.

4.1.1 Centroid Parameters

On each dimension of a multinomial component, the probabilities of all possible category values sum up to 1. Mathematically, that means $\sum_{h=1}^M \mu_k^{jh} = 1$ holds for $k = 1, \dots, K, j = 1, \dots, D$. We use a Dirichlet prior on each dimension of a multinomial centroid. For the purpose of simplicity and fast convergence of the Markov chain, we fix the parameter of the Dirichlet priors to a constant. For $k = 1, \dots, K, j = 1, \dots, D$,

$$(\mu_k^{j1}, \dots, \mu_k^{jm}) \sim \text{Dir}(1, \dots, 1). \quad (5)$$

Choosing a Dirichlet prior can make the derivations of the posterior distribution of the centroids much easier. For a discrete multinomial distribution on $\{1, \dots, M\}$ defined by $(\mu_k^{j1}, \dots, \mu_k^{jm})$, if:

$$(\beta_k^{j1}, \dots, \beta_k^{jm}) \sim \text{Mult}(\mu_k^{j1}, \dots, \mu_k^{jm}), \quad (6)$$

where β_k^{jh} represents the number of occurrences of category value h on j^{th} dimension of k^{th} cluster, then:

$$(\mu_k^{j1}, \dots, \mu_k^{jm}) \sim \text{Dir}(1 + \beta_k^{j1}, \dots, 1 + \beta_k^{jm}) \quad (7)$$

In each Gibbs sampling sweep, the value of β_k^{jh} is updated accordingly with the current sample of the indicator variable c , which indicates the membership of the observations in the clusters.

4.1.2 Mixing Proportions

In the proposed model, we perform linear regression on the mixing proportions. We assume the time stamps associated with the n observations are labeled from t_b to t_e . As in other models using Dirichlet process mixtures [10], we give Dirichlet priors to the K component weights at the beginning time slice, and at the ending time slice. However, since it is simpler to work with independent random variables, we can make use of a reparameterization where the interdependent random variables from a K -variate Dirichlet distribution are viewed as being generated from a set of K independent Gamma random variables. Formally, if $Y_i, i = \{1, \dots, K\}$

are independent variables with $Y_i \sim G(\text{shape} = \alpha_i, \text{scale} = 1)$, then

$$(X_1, \dots, X_K) = (Y_1/V, \dots, Y_K/V) \sim \text{Dir}(\alpha_1, \dots, \alpha_K),$$

where $V = \sum_{i=1}^K Y_i$.

The mixing proportion $\pi_j, j = 1, \dots, K$ at the time slice t can then be expressed as:

$$\pi_j^{(t)} = \frac{b_j}{\sum \bar{b}} + \frac{t - t_b}{t_e - t_b} \left(\frac{e_j}{\sum \bar{e}} - \frac{b_j}{\sum \bar{b}} \right), \quad (8)$$

where $\bar{b} = \{b_1, \dots, b_K\}$ and $\bar{e} = \{e_1, \dots, e_K\}$ have Gamma prior distributions. Since the derivations associated with \bar{e} are very similar to those of \bar{b} , for the sake of simplicity, we just give the derivations associated with \bar{b} in the following discussions. The prior distribution of \bar{b} is:

$$b_j \sim G(\eta_b, 1) \Rightarrow p(b_j|\eta_b) = b_j^{\eta_b-1} \times \frac{e^{-b_j}}{\Gamma(\eta_b)} \quad (9)$$

Assume a data point y_i has time stamp t . The prior of its corresponding indicator variable c_i is given by

$$p(c_i = k|\bar{b}, \bar{e}) = \pi_k^{(t)}, \quad (10)$$

The posterior for the indicator can be obtained by multiplying the prior from Eqn. (10) by the likelihood from Eqn. (4) conditioned on the indicator:

$$p(c_i = k|.) = \left[\frac{b_k}{\sum \bar{b}} + \frac{t - t_b}{t_e - t_b} \left(\frac{e_k}{\sum \bar{e}} - \frac{b_k}{\sum \bar{b}} \right) \right] \times \left[\prod_{j=1}^D \prod_{h=1}^M (\mu_k^{jh})^{x_i^{jh}} \right] \quad (11)$$

The joint priors of $\{c_1, \dots, c_n\}$ can be computed by taking the product of the distributions of indicator variables, iterating over all

of the time slices:

$$p(c_1, \dots, c_n | \vec{b}, \vec{e}) = \prod_{t=t_b}^{t_e} \prod_{k=1}^K \left(\pi_k^{(t)} \right)^{n_k^{(t)}}, \quad (12)$$

where $n_k^{(t)}$ means the number of data points arriving at time t that belong to the k^{th} cluster.

For the hyperparameter \vec{b} , Eqn. (12) plays the role of the likelihood, which together with the prior from Eqn. (9), give conditional posterior of \vec{b} :

$$p(b_j | \cdot) = b_j^{\eta_b - 1} e^{-b_j} \times \prod_{t=t_b}^{t_e} \prod_{k=1}^K \left[\frac{b_k}{\sum \vec{b}} + \frac{t - t_b}{t_e - t_b} \left(\frac{e_k}{\sum \vec{e}} - \frac{b_k}{\sum \vec{b}} \right) \right]^{n_k^{(t)}} \quad (13)$$

For the hyperparameter η_b in Eqn. (9), we assume the inverse Gamma prior. For simplicity, we use a fixed constant value as the parameter of the prior.

$$\eta_b \sim IG_R(1, 1) \Rightarrow p(\eta_b) \propto \eta_b^{-\frac{3}{2}} \exp\left(-\frac{1}{2\eta_b}\right) \quad (14)$$

For η_b , Eqn. (9) plays the role of the likelihood, which together with the prior from Eqn. (14) gives the conditional posterior:

$$p(\eta_b | \cdot) \propto \eta_b^{-\frac{3}{2}} \exp\left(-\frac{1}{2\eta_b}\right) \times \frac{1}{\Gamma^K(\eta_b)} \prod_{j=1}^K b_j^{\eta_b - 1} \quad (15)$$

4.2 Mixture Model with Gaussian Components

The likelihood of the Gaussian mixture model is similar to Eqn. (4), except that the density of a data point is transformed from multinomial to Gaussian:

$$p(y_i | \vec{\mu}, \vec{s}, \vec{\pi}) = \sum_{j=1}^K \pi_j \mathcal{N}(\mu_j, s_j^{-1}), \quad (16)$$

where \mathcal{N} is a Gaussian distribution with specified centroid and variance. In addition to the centroids $\vec{\mu}$ and mixing proportions $\vec{\pi}$, the Gaussian mixture model has a set of parameters $\vec{s} = \{s_1, \dots, s_k\}$, which represent the precisions (inverse variances) of the Gaussian components.

We refer to [19] and use the prior distributions in that work for centroids, precisions and their associated hyperparameters. Given the priors, we derive the posterior distribution of the parameters by Bayesian rule in Eqn. (3). The derivation process is omitted from this paper since it is similar to that of Section 4.1.2.

As for the mixing proportions of Gaussian components, since the modeling of the weights is independent of the type of the mixture component, the conditional distributions of the mixing proportion-related parameters and hyperparameters are the same as in the multinomial mixture model. The only difference in the Gaussian mixture model is the posterior distribution of the indicator variable. The posterior distribution of c becomes:

$$p(c_i = k | \cdot) = \left[\frac{b_k}{\sum \vec{b}} + \frac{t - t_b}{t_e - t_b} \left(\frac{e_k}{\sum \vec{e}} - \frac{b_k}{\sum \vec{b}} \right) \right] \times s_k^{\frac{1}{2}} \exp\left[-\frac{1}{2} s_k (y_i - \mu_k)^2\right]. \quad (17)$$

The generalization to D -dimensional Gaussian components is based primarily on the scheme proposed in [19]. The centroids μ_j and hyperparameter λ become vectors, and their priors become multivariate Gaussians accordingly.

5. EXPERIMENTS

This section describes two sets of experiments designed to investigate the utility of our new mixture model for learning and exhibiting useful temporal trends in component prevalence in real-life data. In the first set of experiments, we use multivariate multinomial distributions to model the antibiotic resistance patterns of *E. coli* bacteria over time. In the second set of experiments, we use multivariate Gaussian distributions to model the antibiotic resistance patterns of *Staphylococcus aureus* (abbreviated as *S. aureus* in medical literature).

5.1 Experiment One: *E. Coli*

In the first experiment, we apply our model to real-life resistance data describing the resistance profile of *E. coli* isolates collected from a group of hospitals. *E. coli* is a food-borne pathogen and a bacterium that normally resides in the lower intestine of warm-blooded animals. There are hundreds of strains of *E. coli*. Some strains can cause illness such as serious food poisoning in humans. For example, *O157 : H7* is an *E. coli* strain that caused a 2006 United States *E. coli* outbreak related to the consumption of fresh spinach. Most strains of *E. coli* will belong to classes that show common resistance patterns to antimicrobial drugs, which is exactly what our model is designed to detect. For example, some studies [12] show that the resistance pattern of *E. coli* with VTEC-AVF is similar to the resistance pattern of enterohemorrhagic *E. coli* (EHEC) and Verocytotoxin-producing *E. coli* (VTEC). It is also reasonable to expect that the prevalence of strains changes over time, which is again what our model is designed to detect. For example, the first case of *ESBL E. coli* (an *E. coli* strain that produces Extended-Spectrum Beta Lactamase, an enzyme which makes the bacteria resistant to several antibiotic drugs) appeared about four years ago and seemed to infect only elderly women. As this strain has spread over time, the age and type of patient who gets infected are also broadened, and hence the population of this strain expectedly increased.

Given these observations, we would expect that by applying our model to the sort of data described in the Introduction of this paper, we would obtain patterns that would be quite useful in terms of their ability to communicate to an epidemiologist what resistance patterns exist in the data, and how the patterns change over time.

Experimental Setup. The *E. coli* test data takes the format of the data set given in Table 1 in the Introduction. In Experiment One, each data point represents the susceptibility of a single isolate collected at one of several, real-life hospitals. In the test data set, there are 9660 *E. coli* isolates tested against 27 antibiotics. Since this is categorical data, we use a multinomial distribution to model the resistance patterns where each category is ‘‘R’’, ‘‘S’’ or ‘‘U’’ as described in Section 1. The date of the isolates ranges from year 2004 to year 2007. We run 2000 loops of our Gibbs sampler and set the number of mixing components $K = 5$ in the experiment. In real-life application, it is difficult to choose the optimal value of the number of clusters, and this is a research problem of its own [11]. However, in our problem, the choice of the number is more application-oriented and it will not affect the correctness of our model. A larger number of clusters indicates a set of fine-grained trends, and a smaller number indicates coarse-grained trends. To avoid the oscillations of the initial phase, we use the mean value of the last 1500 samples as the approximation of each parameter.

Experimental Results. Given these settings, our algorithms compute five resistance patterns (or cluster centroids) for the various groups of *E. coli*. For a particular pattern on a particular antibiotic, the centroid is a three by 27 matrix, where each column in the ma-

trix represents the probability that an isolate from this pattern falls into the categories of “R”, “S” and “U”, respectively.

To visualize our results graphically, in Figure 6 we plot the probability of “S” for each of the five patterns. The sum of the probabilities of resistance and undetermined can be calculated by $(1 - \text{probability of “S”})$.

Also, the linear trends of the mixing proportions learned by our model for the five resistance patterns are plotted in Figure 7. For example, the mixing proportion of pattern 1 is around 0.1 in year 2004, and increases to around 0.2 in year 2007.

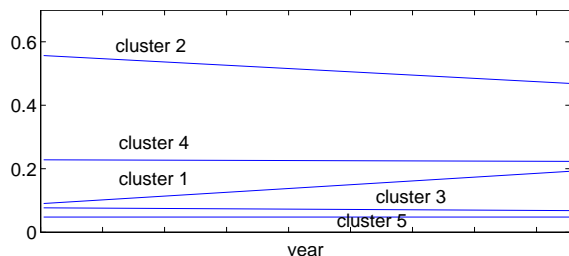


Figure 7: The linear trend of the mixing proportions of the five resistance patterns of *E. coli*. Each line corresponds to a resistance pattern. The start point of a line is the mixing proportion of that pattern at the beginning of year 2004. The end point is the mixing proportion of that pattern at the end of year 2007.

Discussion. The results we observe are quite informative, and also in-keeping with what we might expect to observe in this application domain. For example, consider pattern two. This pattern corresponds to those isolates that are highly susceptible to almost all of the relevant antimicrobials. It turns out that this is also the most prevalent class of *E. coli*, which is very good news. In 2004, more than 55% of the isolates belonged to this class. Unfortunately, presumably due to selective pressures, the prevalence of this class decreases over time. The learned model shows that by 2007, the prevalence of the class had decreased to around 45%. This sort of trend—a decrease in prominence of a specific pattern—is exactly what our model is designed to detect.

While the decrease in prevalence of pattern two is worrisome, there is some good news from the data: the prevalence of patterns three and five, which correspond to *E. coli* that shows the broadest antimicrobial resistance, generally does not change over time, and is rather flat.

Clusters three and five are actually quite interesting, because from a clinical standpoint there seems to be a clear reason to think that there would be natural movement from pattern one to pattern four, then to pattern three, and then into pattern five. The reason is that the so-called Cephalosporins—one of the key drug classes for treatment of *E. coli*—can be categorized into “generations” based upon when they were introduced. Cefazolin and Cefuroxime are early-generation Cephalosporins; in pattern one, *E. coli* is just beginning to show resistance to these drugs, with a susceptibility of 92% to Cefazolin. However, in pattern four, susceptibility drops to 88%, and it drops to 13% in pattern three and 0% in pattern five. *E. coli* in pattern one and pattern four has no resistance at all to any of the so-called third-generation Cephalosporins—Ceftriaxone, Cefotaxime, and Ceftriaxime—but it has some limited resistance in pattern three and significant resistance in pattern five. The most worrisome aspect of pattern five is that *E. coli* in this class seems to have evolved the presence of ESBL, which causes resistance to advanced-generation Cephalosporins, of which Ceftriaxone, Cefotaxime, and Ceftazidime (drugs 12, 15, and 16) are all members. Pattern five is the only pattern with any resistance to these drugs.

While patterns three, four, and five do not seem to change in

prevalence, the one significant movement is the increase in the prevalence of pattern one over time; one might infer from the learned model that there is evolution in the population of *E. coli* from pattern two (highly susceptible *E. coli*) into pattern one. Given the natural progression of Cephalosporin resistance from the first generation through the advanced drugs such as Ceftriaxone, this is cause for some concern. If these bugs continue the natural progression in Cephalosporin resistance, one might expect that the increase in pattern one’s prevalence portends a future increase in the prevalence of pattern four, then three, and then eventually pattern five.

It is also particularly interesting that pattern one would grow significantly in prevalence, since this class of *E. coli* shows significant resistance to Levofloxacin, Ciprofloxacin, and Moxifloxacin (drugs 11, 13, and 14). These three antimicrobials all belong to the class of drugs called Fluoroquinolones. These are broad-spectrum antimicrobials that are among the most overused/abused in the last decade, and hence there is significant pressure on bacteria to produce resistance to the drugs, which they certainly seem to do. In fact, it seems that resistance to these drugs is the first sort of resistance developed by sub-populations of *E. coli*.

5.2 Experiment Two: *S. Aureus* Bacteria

The test data in Experiment Two are collected by the Antimicrobial Resistance Management (ARM) Program [1], which is an ongoing project designed to document trends in antimicrobial susceptibility patterns in inpatient and outpatient isolates, and to identify relationships between antibiotic use and resistance rates. The data give the susceptibility of 19 bacteria to 51 antibiotics collected from 355 participating US hospitals. Each hospital provides a minimum of three years of susceptibility data.

In the experiment, we investigate the resistance patterns of *S. aureus* in the ARM data. *S. aureus* is known to have various subclasses or strains. Most hospital strains of *S. aureus* are usually resistant to a variety of different antibiotics, and a few strains are resistant to all clinically useful antibiotics except Vancomycin, and Vancomycin-resistant strains are increasingly reported. The prevalence of another strain, named Methicillin Resistant *Staphylococcus aureus* (MRSA), is widespread too.

Experiment Setup. Unlike in Experiment One, where each data point represents the susceptibility of a single isolate, in Experiment Two, the susceptibility data is aggregated based on all the bacterial isolates collected from a given hospital in a particular year. Each data point is then a susceptibility rate vector, associated with a specific hospital and a specific year. The susceptibility rate vector consists of M real numbers, where each number represents the fraction of bacteria isolates collected in a given year that are susceptible to the M antibiotics. In order to make the experiment simple and informative, we model resistance to a set of 17 clinically relevant antibiotics. The time span we tested is from year 1992 to year 2006 and the data set size is 1323. Since the susceptibility rates are vectors of real numbers, we model them using multivariate Gaussian distributions. We set $K = 3$, and run 2000 loops of Gibbs sampler in the experiment. As in Experiment One, we use the mean value of the last 1500 samples as the approximate value of each parameter.

Experimental Results. Figure 8 gives three different clusters for *S. aureus*, each of which is associated with a specific resistance patterns. All the patterns of *S. aureus* have susceptibility rates larger than 0.5. Pattern 1 has the highest susceptibility rates over a large portion of antibiotics. The susceptibility rates of the other two patterns are generally lower except for Cefazolin, Linezolid, Meropenem, and Quinupristin/Dalfopristin.

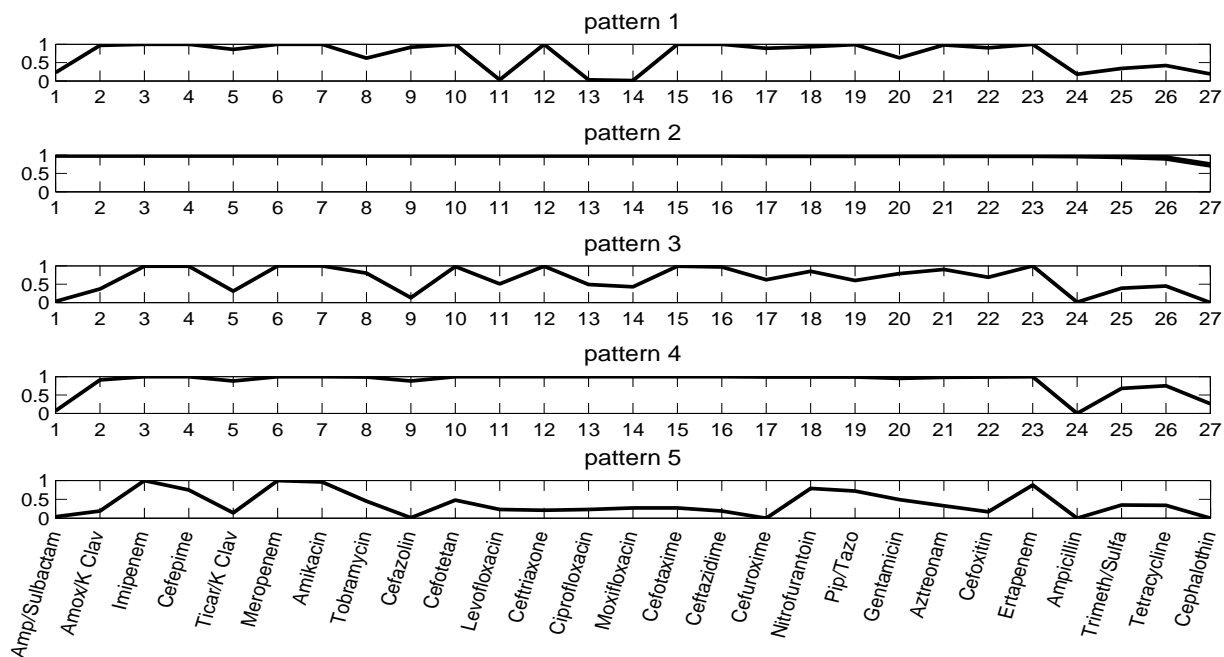


Figure 6: The susceptibility of *E. coli* to multiple antibiotics. The X-axis represents antibiotic names. The Y-axis represents the susceptible probability. Each curve represents a resistance pattern. For example, in pattern 5, an *E. coli* isolate is susceptible to Imipenem with a probability of 1.

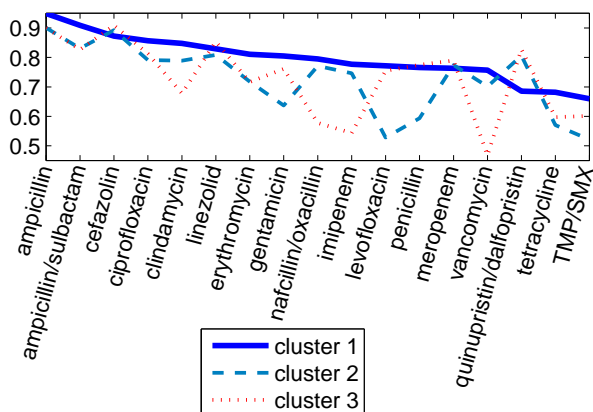


Figure 8: The susceptible rate of the three resistance patterns of *S. aureus*. Each curve represents a resistance pattern. For example, in pattern three, 67% of isolates are susceptible to Clindamycin.

The linear trends of the mixing proportions of the three resistance patterns exhibited by *S. aureus* are plotted in Figure 9. Pattern one occupies a dominant portion of nearly 100% at the beginning of time. At the end of the time, although pattern one is still the most prevalent pattern, its dominance is diminished and patterns two and three gain more prevalence over time.

Discussion. Unlike in the first experiment, in this experiment the learned patterns do not correspond to strains of microbes, they correspond to resistance profiles for *hospitals*. The results show that the “best” profile (where the highest susceptibility rates are found) decreases significantly over time; this is expected, and not good news. Of epidemiological interest is the increase in prevalence of pattern three over time. This pattern shows a high rate of isolates

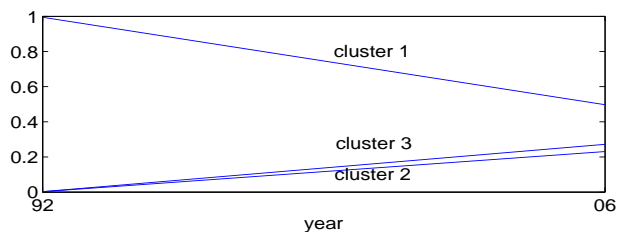


Figure 9: The linear trend of the mixing proportions of the three resistance patterns of *S. aureus*. The two ends of a line represent the mixing proportion of the pattern in year 1992 and year 2006 respectively.

that are not susceptible to Clindamycin, which is indicative of a high rate of MRSA, a worrisome strain of *S. aureus*.

One interesting finding is that both patterns two and three are marked by a very high *susceptibility* to the Quinupristin/Dalfopristin combination. This is quite interesting, because it means that Quinupristin/Dalfopristin susceptibility is associated with hospitals showing *S. aureus* with a *lack* of susceptibility to other drugs, and that such hospitals are becoming *more* common over time. We were so startled by this observation that we tried re-running the learning algorithm with a greater value for K , since we thought that one explanation could be that several hospital profiles were being mixed together within each pattern. However, increasing K had little effect: all it did was to split up patterns two and three many ways, and each pattern other than one *still* had a very high rate of susceptibility to Quinupristin/Dalfopristin!

6. RELATED WORK

Mining of temporal data has received significant attention in the literature. It is not feasible to provide a detailed overview of related work because of space limitations. In this section, we describe the research most closely related to our own work.

In time-series data mining [3, 8, 16], each data point typically belongs to one of k labeled time series that are archived in the database—for example, a database may contain ECG data for k different people. All time series are generally independent, but within each time series there is an implicit (or explicit; see [3]) assumption that the time series is generated from a data generator with internal state. This is fundamentally different from our setup where there is a single time series with no internal state; all data are i.i.d. samples from a single, time-evolving model which we try to discover.

The closest work to our own is concerned with mining document features or document classes that evolve over time [4, 17, 21]. The document classes that are mined can be seen as equivalent to our cluster centroids instantiated using a “bag of words” model (see, for example, Blei, Ng, and Jordan [5]). The key feature that differentiates our work is that our classes are fixed; what changes over time is only the prevalence of each class, and this change is modeled as a simple, linear progression. This results in a simple, easy-to-understand model, where (with enough data) the learned model is invariant to the distribution of the time attribute itself—an issue that we discussed in detail in Section 2.

Much work has been done in clustering temporal data. Some clustering methods discretize the data based upon time [14, 2]. In these methods, one model is built per time period, or the model is updated to incorporate the new data as the “recent” time window moves forward. Some other time-based clustering methods are concerned with maintaining temporal “smoothness” [6, 7]. Methods utilizing temporal smoothness attempt to fit the current data well, while at the same time, avoiding too much deviation from the historical clustering.

Finally, temporal anomaly detection is also somewhat related to our work [18, 15]. In temporal anomaly detection, the goal is to find anomalous, emergent patterns. This could be seen as discovering classes that emerge suddenly, and grow from a prevalence of zero in a short time.

7. CONCLUSION AND FUTURE WORK

We have developed a Bayesian mixture model where the mixing proportions of the components change over time, and evaluated the model by applying it to the real-life antibiotic resistance data.

There are many potential avenues for future work. Consider the application to the ARM database in Section 5.2. In this case, it is actually known before our algorithms are applied which susceptibility vectors correspond to which hospitals, so each hospital in reality forms an individual time series. Currently, our algorithms ignore this information and simply mix all hospitals together, then learn trends from the mixed data. However, such labels could be quite useful, because they could be used to help us learn how the different hospitals transition between clusters over time. For example, we might be able to learn that hospitals start in Cluster 1, then move to Cluster 2, then move to Cluster 3 over time, as opposed to simply learning the change in cluster weight over time, as our algorithms do now. This could be quite informative to an analyst, because it not only shows how the weight changes, but it shows how individual time series evolve.

Finally, we mention that (as pointed out by one of the anonymous reviewers of this paper), we have not yet conducted a rigorous study of the computational and statistical properties of the MCMC algorithms derived in the paper. For example, it would be useful to know how quickly our algorithms tend to “mix”—that is, given a reasonable application domain and model, how quickly the Markov chain reaches a steady state where it repeatedly samples from the true, posterior distribution of the model given the data.

8. ACKNOWLEDGEMENTS

Material in this paper is based upon research supported by the National Science Foundation under grant number 0612170.

9. REFERENCES

- [1] <http://www.armprogram.com/>.
- [2] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *VLDB*, pages 81–92, 2003.
- [3] A. J. Bagnall and G. J. Janacek. Clustering time series from arma models with clipped data. In *KDD*, pages 49–58, 2004.
- [4] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [6] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *KDD*, pages 554–560, 2006.
- [7] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *KDD*, pages 153–162, 2007.
- [8] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *KDD*, pages 493–498, 2003.
- [9] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Stat. Soc., Series B*, 39(1):1–38, 1977.
- [10] T. S. Ferguson. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, vol. 1, no. 2:209–230, 1973.
- [11] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):381–396, 2002.
- [12] K. G. and B. M. Antibiotic susceptibility pattern of escherichia coli strains with verocytotoxic e. coli-associated virulence factors from food and animal faeces. *Food Microbiology*, Volume 20, Number 1:27–33(7), February 2003.
- [13] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution and bayesian restoration of images. *IEEE PAMI*, vol. 6:721–741, 1984.
- [14] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *KDD*, pages 97–106, 2001.
- [15] A. T. Ihler, J. Hutchins, and P. Smyth. Adaptive event detection with time-varying poisson processes. In *KDD*, pages 207–216, 2006.
- [16] E. J. Keogh, S. Lonardi, and B. Y. chi Chiu. Finding surprising patterns in a time series database in linear time and space. In *KDD*, pages 550–556, 2002.
- [17] J. M. Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7(4):373–397, 2003.
- [18] D. B. Neill, A. W. Moore, M. Sabhnani, and K. Daniel. Detection of emerging space-time clusters. In *KDD*, pages 218–227, 2005.
- [19] C. E. Rasmussen. The infinite gaussian mixture model. In *NIPS*, pages 554–560, 1999.
- [20] P. Schlattmann. Estimating the number of components in a finite mixture model: the special case of homogeneity. *Comput. Stat. Data Anal.*, 41(3-4):441–451, 2003.
- [21] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD*, pages 424–433, 2006.