A robust hidden semi-Markov model with application to aCGH data processing

Jiarui Ding

Department of Computer Science,
University of British Columbia

and

Department of Molecular Biology,
BC Cancer Agency,
Vancouver, BC, V5T 4E6, Canada
E-mail: jiaruid@cs.ubc.ca

Sohrab Shah*

Department of Computer Science,
University of British Columbia

and

Department of Molecular Biology,
BC Cancer Agency

and

Department of Pathology,
University of British Columbia
Vancouver, BC,
V5T 4E6, Canada
E-mail: sshah@bccrc.ca
*Corresponding authors

Abstract: Hidden semi-Markov models are effective at modelling sequences with succession of homogenous zones by choosing appropriate state duration distributions. To compensate for model mis-specification and provide protection against outliers, we design a robust hidden semi-Markov model with Student's t mixture models as the emission distributions. The proposed approach is used to model array based comparative genomic hybridization data. Experiments conducted on the benchmark data from the Coriell cell lines, and glioblastoma multiforme data illustrate the reliability of the technique.

Keywords: array CGH data; copy number variation; hidden semi-Markov models; discriminative training; Student's t distribution; rhsmm.

Reference to this paper should be made as follows: Ding, J. and Shah, S. (xxxx) 'A robust hidden semi-Markov model with application to aCGH data processing', *Int. J. Data Mining and Bioinformatics*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Jiarui Ding is a PhD candidate in the Department of Computer Science at the University of British Columbia, and Department of Molecular Biology in British Columbia Cancer Agency, Canada. He received his MSc Degree in Mechanical Engineering from the University of Saskatchewan, Canada, in 2009. His research interests include machine learning and its applications to bioinformatics. Currently he is working on somatic genomic variation prediction form next generation sequencing data, and causal driver gene prediction in cancer.

Sohrab Shah is an Assistant Professor in the Department of Pathology at UBC and a Scientist at the British Columbia Cancer Agency. He is also an Associate Member of the Department of Computer Science at UBC. He obtained his PhD in Computer Science from UBC working with Drs. Kevin Murphy and Raymond Ng. He then completed a post-doctoral fellowship with Drs. David Huntsman and Sam Aparicio at the BC Cancer Agency. He work in breast and ovarian cancer and lymphoma has led to discoveries that were published in journals such as Nature and the New England Journal of Medicine.

1 Introduction

Machine learning techniques have been widely used in science and engineering. Here we consider probabilistic approaches. Assume that a probabilistic classifier with parameter vector θ, a discriminative classifier models $p(c|x, \theta)$ directly, where c is a class label and x is a feature vector. If a classifier models the joint probability $p(c, x)$, the classifier is called a generative classifier because random instances of x can be generated by first sampling from $p(c)$, and then sampling from $p(x|c)$.

Generative classifiers have several advantages over discriminative classifiers. For example, generative classifiers are easy to fit, most times just by counting and average, based on maximum likelihood estimation

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=0}^{N-1} \log p(c_i, x_i|\theta).$$

where $\{x_0, ..., x_{N-1}\}$ are the training data. These models can handle unlabeled data. This property makes them very attractive in semi-supervised and un-supervised learning. However, one disadvantage of generative models is that they may be less accurate than discriminative models. The reason behind is that the conditional density $p(x|c, \theta)$ may be high-dimensional complex distributions. These distributions may be difficult to be approximated by parametric models. Therefore, model mis-specification may degrade the performance of generative classifiers.

To get better performance, generative models can be trained discriminatively given a labelled training set. The fundamental idea behind discriminative training is to consider the inaccuracy of the model, and directly optimise the misclassification errors. For example, we can estimate θ by

$$\arg\max_{\theta} \sum_{i=0}^{N-1} \log p(c_i|x_i, \theta) = \arg\max_{\theta} \sum_{i=0}^{N-1} \log \frac{p(x_i, c_i|\theta)}{\sum_c p(x_i, c|\theta)}.$$

However, discriminative training is time consuming and we need to choose the learning rates (Cheng et al., 2010). Another way to consider the inaccuracy of models is to use robust methods (Peel and McLachlan, 2000). These robust methods may get even better performance than discriminative training methods but keep the simplicity of maximum likelihood estimators (Chatzis et al., 2009).

Hidden Markov Models (HMM) are generative classifiers. Discriminative training of HMMs has a long history, especially in the automatic speech recognition area (Cheng et al., 2010). Of these discriminative trained models, recently, the large-margin HMMs show better performance. However, discriminative training of HMMs is more complicated than maximum likelihood estimation because discriminative training methods should separate correct hypothesis from incorrect ones (Cheng et al., 2010). The Expectation-Maximisation (EM) algorithm can not be used for discriminative training and genetic optimisation algorithms should be used. Recently, the robust HMMs which use mixture of Student's t distributions as the emission distributions show promising results (Chatzis et al., 2009).

One disadvantage of HMMs is that the probability distribution of staying in the same state is a geometric distribution. For many real-world problems, this is not the case. One solution is to explicitly model the state duration distribution. This more powerful model which explicitly models the state duration distributions is called a Hidden Semi-Markov Model (HSMM), which can better model the sequences with succession of homogenous zones. A problem of HSMMs is that the computational complexity of inference and learning algorithms is very high. In 2003, an efficient inference and learning algorithm were introduced (Guedon, 2003). However, these algorithms only consider discrete emission distributions and the applications of these algorithms are limited.

In this paper, we extend the robust HMM to robust HSMMs (Yu, 2010), and apply the robust HSMMs to analyse array based Comparative Genomic Hybridisation (aCGH) data. Because HMMs are special cases of HSMMs, our models are more general and can be used to model a large number of sequential data. These robust methods may achieve even better performance than discriminative training without increasing the training complexity, and can be used with or without labelled training data.

2 Hidden Semi-Markov Models

The HSMM is a generative model which defines a joint distribution over the discrete state sequence $\{S_t\}$ and discrete or continuous observation sequence $\{X_t\}$. The discrete state sequence is modelled by a semi-Markov chain, which means to predict the next state, we need to know not only the current state, but also how long we have been in this state. Condition on the current state, the current observation is independent of other states. Because the state space is not observable directly, the model is called a hidden semi-Markov model.

To write down the joint distribution of HSMMs, firstly, we define some variables. The observation sequence is represented as $x_0^{\tau-1} : x_0 \ldots, x_{\tau-1}$ and the hidden sequence is represented as $s_0^{\tau-1} : s_0 \ldots, s_{\tau-1}$, where τ is the length of the sequence. The initial probability is given by $\pi_j = P(S_0 = j)$ with $\sum_{j=0}^{J-1} \pi_j = 1$ where J is the number of hidden states. For a non-absorbing state i, the translation probability is given by

$$\begin{cases} A_{ij} = P(S_{t+1} = j \mid S_t = i), \sum_j A_{ij} = 1 \; i \neq j; \\ A_{ii} = P(S_{t+1} = i \mid S_t = i) = 0, \qquad . \end{cases}$$

Here $A_{ii} = 0$ because we already explicitly model the state duration distribution. For an absorbing state I

$$A_{ii} = P(S_{t+1} = i \mid S_t = i) = 1$$

We also define the observation probability of state j given observation x_t

$$e_j(x_t) = P(X_t = x_t \mid S_t = j)$$

A state duration (sojourn time, occupation) distribution d_j (u) is attached to each non-absorbing state j.

$$d_j(u) = \begin{cases} P(S_{t+u+1} \neq j, S_{t+2}^{t+u} = j \mid S_{t+1} = j, S_t \neq j), \\ \qquad\qquad\qquad\qquad u = 2, \ldots, M_j; \\ P(S_{t+2} \neq j \mid S_{t+1} = j, S_t \neq j), \qquad u = 1. \end{cases}$$

Here we define $S_{-1} = -1$ to make the above formula true at the beginning. $d_j(u)$ is the probability of staying in state j for only u steps. For $d_j(u)$, we only consider discrete distributions here. M_j is the upper bound of the time spent in state j. Here M_j is used for the definition of right censoring, which means the end of a sequence may not coincide with the exiting of a state. The survivor function

$$D_j(u) = \sum_{u \leq v \leq M_j} d_j(v)$$

represents the marginal time spent in state j by summing over all the possible time longer that or equal to u. The normalisation term is defined as

$$N_t = P(X_t = x_t \mid X_0^{t-1} = x_0^{t-1})$$

It is the normalisation term N_t preventing numerical underflow, and moreover, the products of N_t is the likelihood, which can be used to check the correctness of our code (the likelihood should monotonically increase for the EM algorithm (Bilmes, 1997).

Figure 1 shows an simple HSMM with three states. Note the state duration distribution $d_j(u)$ can be arbitrary distributions. While for HMM, $d_j(u)$ is constrained to be a geometric distribution. For example, given a hidden state sequence $s_0^7 = 00011222$, and the observation sequence x_0^7, the join distribution is as:

$$P(s_0^7, x_0^7) = \pi_0 d_0(3) A_{01} d_1(2) A_{12} d_2(3) \prod_{t=0}^{7} b_{s_t}(x_t)$$

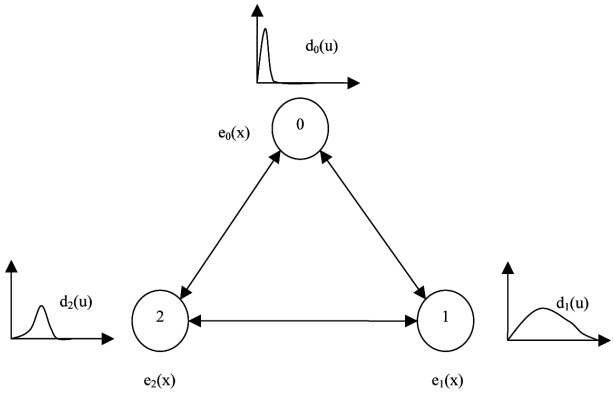
if we don't consider censoring.

As in Guedon (2003), we assume the observation sequences are right censored. Given an observation sequence $x_0^{\tau-1}$, the underlining state sequence is $s_0^{\tau-1+u}$. The extra states $s_\tau^{\tau-1+u}$ are introduced to give a complete state sequence up to the exit from the state $s_{\tau-1}$. The joint distribution can be written as follows:

$$
\begin{aligned}
&f(s_0^{\tau-1+u}, x_0^{\tau-1}) \\
=&P(S_0^{\tau-1} = s_0^{\tau-1}, S_\tau^{\tau+u-2} = s_{\tau-1}, S_{\tau-1+u} \neq s_{\tau-1}, x_0^{\tau-1}) \\
=&\pi_{s_0} d_{s_0}(u_0) \prod_{r=1}^{R} p_{s_{(\eta_r-1)} s_{\eta_r}} d_{s_{\eta_r}}(u_r) \\
&\prod_{t=0}^{\tau-1} e_{s_t}(x_t) I(\eta_R < \tau - 1 \leq \eta_{(R+1)})
\end{aligned}
$$

where $\eta_r \sum_{i=0}^{r-1} u_i, 1 \leq r \leq R+1$, 1 < r < R + 1 and R is the number of state translations.

Figure 1 Graphical representation of an HSMM with 3 hidden states



Because the state sequence is not known, we use the EM algorithm for parameter estimation (Bilmes, 1997). In the E-step, the expectation of the complete log likelihood function is computed as follows

$$
\begin{aligned}
Q(\theta \mid \theta^{(i)}) = &\sum_{u} \sum_{s_0}^{s_{\tau-1+u}} \log f(s_0^{\tau-1+u}, x_0^{\tau-1}) \\
&* p(s_0^{\tau-1+u} \mid x_0^{\tau-1}, \theta^{(i)})
\end{aligned} \tag{1}
$$

As in Bilmes (2007) and Guedon (2003), the Q function can be partitioned into four groups according to the parameters π_i, A_{ij}, $d_j(u)$ and $e_j(x_t)$. For simplicity, we only give the Q function for $e_j(x_t)$. Other Q functions can be deduced similarly.

$$
\begin{aligned}
Q(e|\theta^{(i)}) &= \sum_{u} \sum_{s_0}^{s_{\tau-1+u}} \sum_{t=0}^{\tau-1} \log(e_{s_t}(x_t)) p(s_0^{\tau-1+u} | x_0^{\tau-1}, \theta^{(i)}) \\
&= \sum_{j=0}^{J-1} \sum_{t=0}^{\tau-1} \log(e_j(x_t)) p(s_t = j | x_0^{\tau-1}, \theta^{(i)}) \\
&= \sum_{j=0}^{J-1} \sum_{t=0}^{\tau-1} \log(e_j(x_t)) L_j(t)
\end{aligned} \tag{2}
$$

The first '=' in equation (2) is true because of the Markovian property. The second '=' is true because for the parameter $e_j(x_t)$, we only care about the current state according to the

Markovian property. In other words, we need to marginalise the posterior $p(s_0^{\tau-1+u} \mid x_0^{\tau-1}, \theta^{(i)})$ to get $p(s_t = j \mid x_0^{\tau-1}, \theta^{(i)})$.

2.1 Emission probability distribution

The emission probability may be modelled by a large number of distributions. The Gaussian Mixture Models (GMM) are widely used because they can approximate any distribution as the number of mixture components increases. However, GMMs are sensitive to outliers, and some extra mixture components are needed to model these outliers. To increase the number of mixture components may not be a good idea because the time, the number of data should also increase to fit the model. The Student's t distribution has longer tails and provides a robust version of the Gaussian distribution. In this paper, we will use the Student's t mixture model (SMM) as the emission distribution.

The SMM is defined as

$$p(x) = \sum_{k=0}^{K-1} p(Z = k)\mathcal{T}(x \mid Z = k, \mu_k, \Sigma_k, \nu_k) \tag{3}$$

where K is the number of mixture components, $p(Z = k) = c_k$ is the mixing weight which satisfies $0 \leq < c_k \leq 1$ and $\sum_k c_k = 1,$ and the distribution

$$\mathcal{T}(x \mid Z = k, \mu_k, \Sigma_k, \nu_k)$$
$$= \frac{\Gamma(\frac{\nu_k + \tilde{D}}{2})}{\Gamma(\frac{\nu_k}{2})} \frac{|\Sigma_k|^{-1/2}}{(\pi\nu_k)^{\frac{1}{2}\tilde{d}}} (1 + \frac{\Delta_x^2(\mu_k, \Sigma_k)}{\nu_k})^{-\frac{1}{2}(\nu_k + \tilde{D})}$$

where $\Gamma(x) = \int_0^\infty v^{x-1} e^{-v} \mathrm{d}v$ and $\tilde{D}$ is the dimensionality of x. v_k is the degree-of-freedom (dof) which controls the length of the tail. As $v_k \to \infty$, the corresponding Student's t distribution becomes a Gaussian distribution with mean of μ_k and covariance of Σ_k. $\Delta_x^2(\mu_k, \Sigma_k)$ is the square Mahalanobis distance defined by (Bishop, 2006)

$$\Delta_x^2(\mu_k, \Sigma_k) = (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)$$

Now the observation distribution of equation (2) becomes

$$e_j(x_t) = \sum_{k=0}^{K-1} p(Z_{jt} = k)\mathcal{T}(x_t \mid Z_{jt} = k, \mu_{jk}, \Sigma_{jk}, \nu_{jk})$$

where $p(Z_{jt} = k) = c_{jk}$ is the mixing weight. Note that there are no close-form solutions for the degree of freedom V_{jk}. However, we can optimise equation (2) with regard to V_{jk} using the most recent values of c_{jk} and μ_{jk}. This is not difficult because this is only a one dimensional optimisation problem given the value of v_{jk} between (0.1,1000) is sufficient most times.

To get close-form solutions of and μ_{jk}, we turn into another representation of the Student's t distribution as follows:

$$\mathcal{T}(x \mid \mu_k, \Sigma_k, \nu_k) = \int_0^\infty \mathcal{N}(x|\mu_k, \frac{\Sigma_k}{\lambda})\mathrm{Ga}(\lambda \mid \frac{\nu_k}{2}, \frac{\nu_k}{2})d_\lambda$$

where $N\left(x \mid \mu_k, \frac{\Sigma k}{\lambda}\right)$ is a Gaussian distribution with mean of μ_k and covariance of $\Sigma k/\lambda$, and Ga() is the Gamma probability distribution function. This model is called a Gaussian scale mixture model, which is an infinite weighted sum of Gaussians with different precisions (Murphy, In preparation).

As GMMs, the SMMs are generative models. Given a training sample x, we can assume that x is generated as follows: firstly, a number k which specifies the mixture component is generated from $p(z)$, then λ which controls the covariance of the Gaussian component is generated from $\mathrm{Ga}\left(\lambda \mid \frac{vk}{2}, \frac{vk}{2}\right)$, and at last, the sample x is generated from the Gaussian distribution $\mathcal{N}\left(x \mid \mu k, \frac{\Sigma k}{\lambda}\right)$.

Based on the above analysis, we should introduce some extra variables, i.e., a variable Z_{jt} indicating the mixture component for each state j at each time t and the variable A_{jkt} introduced by the Gaussian scale mixture model. Now we can rewrite the Q function as follows:

$$\begin{aligned} Q(e|\theta^{(i)}) &= \sum_{j=0}^{J-1}\sum_{t=0}^{\tau-1}\sum_{k=0}^{K-1} p(S_t = j, Z_{jt} = k, \Lambda_{jkt} = \lambda_{jkt} \\ &\quad |x_0^{\tau-1}, \theta^{(i)}) \log(c_{jk}\mathcal{T}(x_t|Z_{jt} = k, \mu_{jk}, \Sigma_{jk}, \nu_{jk})) \\ &p(S_t = j, Z_{jt} = k, \Lambda_{jkt} = \lambda_{jkt}|x_0^{\tau-1}, \theta^{(i)}) \\ &\quad = p(S_t = j|x_0^{\tau-1}, \theta^{(i)})p(Z_{jt} = k|S_t = j, x_0^{\tau-1}, \theta^{(i)}) \\ &\quad p(\Lambda_{jkt} = \lambda_{jkt}|Z_{jt} = k, S_t = j, x_0^{\tau-1}, \theta^{(i)}) \end{aligned} \tag{4}$$

Based on the Gaussian scale mixture model, the term $T(x_t \mid \mu_{jk}, \Sigma_{jk}, v_{jk})$ can be expanded as follows:

$$\begin{aligned} \log \mathcal{T}(x_t|\mu_{jk}, \Sigma_{jk}, \nu_{jk}) &= \log \mathcal{N}(x_t \mid \mu_{jk}, \frac{\Sigma_{jk}}{\lambda_{jkt}}) \\ &\quad + \log \mathrm{Ga}(\lambda_{jkt}|\frac{\nu_{jk}}{2}, \frac{\nu_{jk}}{2}) \\ \log \mathcal{N}(x_t \mid \mu_{jk}, \frac{\Sigma_{jk}}{\lambda_{jkt}}) &= \frac{\tilde{D}}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_{jk}| \\ &\quad + \frac{\tilde{D}}{2}\log(\lambda_{jkt}) - \frac{1}{2}\lambda_{jkt}\Delta_{x_t}^2(\mu_{jk}, \Sigma_{jk}) \end{aligned} \tag{5}$$

$$\begin{aligned} \log \mathrm{Ga}(\lambda_{jkt}|\frac{\nu_{jk}}{2}, \frac{\nu_{jk}}{2}) &= -\log\Gamma(\frac{\nu_{jk}}{2}) + \frac{\nu_{jk}}{2}\log(\frac{\nu_{jk}}{2}) \\ &\quad - \frac{\nu_{jk}}{2}\lambda_{jkt} + (\frac{\nu_{jk}}{2} - 1)\log(\lambda_{jkt}) \end{aligned} \tag{6}$$

After plugging these terms into the expectation of the complete log likelihood function (4), we need to compute the posterior

$$p(S_t = j|x_0^{\tau-1}, \theta^{(i)}) = L_j(t) \tag{7}$$

$$\begin{aligned} &p(Z_{jt} = k|S_t = j, x_0^{\tau-1}, \theta^{(i)}) \\ &= \frac{p(x_0^{\tau-1}|Z_{jt} = k, S_t = j, \theta^{(i)})p(Z_{jt} = k|S_t = j, \theta^{(i)})}{\sum_k p(x_0^{\tau-1}|Z_{jt} = k, S_t = j, \theta^{(i)})p(Z_{jt} = k|S_t = j, \theta^{(i)})} \\ &= \frac{c_{jk}^{(i)}\mathcal{T}(x_t \mid \mu_{jk}^{(i)}, \Sigma_{jk}^{(i)}, \nu_{jk}^{(i)})}{\sum_{k=0}^{K-1} c_{jk}^{(i)}\mathcal{T}(x_t \mid \mu_{jk}^{(i)}, \Sigma_{jk}^{(i)}, \nu_{jk}^{(i)})} \end{aligned} \tag{8}$$

$$\begin{aligned} &p(\Lambda_{jkt} = \lambda_{jkt}|Z_{jt} = k, S_t = j, x_0^{\tau-1}, \theta^{(i)}) \\ &\quad = \mathrm{Ga}(\frac{v_{jk}^{(i)} + \tilde{D}}{2}, \frac{v_{jk}^{(i)} + \Delta_{x_t}^2(\mu_{jk}^{(i)}, \Sigma_{jk}^{(i)})}{2}) \end{aligned} \tag{9}$$

From equations (5) and (6), we also need to compute the expectations of the hidden variable A_{jkt} and $\log(\Lambda_{jkt})$. Because the posterior of Λ_{jtk} is a Gamma distribution, the expectation

$$\begin{aligned} \tilde{\lambda}_{jkt} &= E(\Lambda_{jkt}|Z_{jt} = k, S_t = j, x_0^{\tau-1}, \theta^{(i)}) \\ &= \frac{\nu_{jk}^{(i)} + \tilde{D}}{\nu_{jk}^{(i)} + \Delta_{x_t}^2(\mu_{jk}^{(i)}, \Sigma_{jk}^{(i)}))} \end{aligned} \tag{10}$$

However, the expectation of $\log(\Lambda_{jkt})$ only influences the estimation of v_{jk}, while v_{jk} can be estimated by optimising equation (2). Therefore, there is no need to compute this term any more.

The E-step is to compute equations (7)–(10). If we define the following term:

$$\tilde{L}_{jk}(t) = \frac{L_j(t)c_{jk}^{(i)}\mathcal{T}(x_t \mid \mu_{jk}^{(i)}, \Sigma_{jk}^{(i)}, \nu_{jk}^{(i)})}{\sum_{k=0}^{K-1} c_{jk}^{(i)}\mathcal{T}(x_t \mid \mu_{jk}^{(i)}, \Sigma_{jk}^{(i)}, \nu_{jk}^{(i)})}$$

By taking the partial derivatives of equation (4) with respect to the parameters and setting to zero to solve a series of equations to get the following M-step:

$$\begin{aligned} c_{jk}^{(i+1)} &= \frac{\sum_{t=0}^{\tau-1} \tilde{L}_{jk}(t)}{\sum_{t=0}^{\tau-1} L_j(t)} \\ \mu_{jk}^{(i+1)} &= \frac{\sum_{t=0}^{\tau-1} \tilde{L}_{jk}(t)\tilde{\lambda}_{jkt}x_t}{\sum_{t=0}^{\tau-1} \tilde{L}_{jk}(t)\tilde{\lambda}_{jkt}} \\ \Sigma_{jk}^{(i+1)} &= \frac{\sum_{t=0}^{\tau-1} \tilde{L}_{jk}(t)\tilde{\lambda}_{jkt}(x_t - \mu_{jk}^{(i+1)})(x_t - \mu_{jk}^{(i+1)})}{\sum_{t=0}^{\tau-1} \tilde{L}_{jk}(t)} \end{aligned}$$

In the next section, we will see how to compute $L_j(t)$, and other necessary terms to estimate the parameters of an HSMM.

2.2 The forward-backward algorithm

Here we briefly describe the forward-backward algorithm. More detail descriptions of the algorithms can be found in Guedon (2003). Note in this section the superscript (i) is

ignored for simplicity. The forward algorithm is to recursively compute the term $F_j(t)$, for $t = 0,\ldots, \tau-2, j = 0,\ldots, J-1$

$$\begin{aligned} F_j(t) =& P(S_{t+1} \neq j, S_t = j \mid X_0^t = x_0^t) \\ =& \sum_{u=1}^{t} \sum_{i \neq j} P(S_{t+1} \neq j, S_{t-u+1}^t = j, S_{t-u} = i \mid x_0^t) \\ & + P(S_{t+1} \neq j, S_0^t = j \mid x_0^t) \\ =& \sum_{u=1}^{t} \sum_{i \neq j} \prod_{v=t-u+1}^{t} \frac{e_j(x_v)}{N_v} * d_j(u) * A_{ij} * F_i(t-u) + \\ & \prod_{v=0}^{t} \frac{e_j(x_v)}{N_v} d_j(t+1)\pi_j \end{aligned} \tag{11}$$

At time step $\tau-1$, because we do not know the exact time spent in that state, only the minimum time spent in the state is known (right censored). We replace $d_j(u)$ by $D_j(u)$ in equation (11) to get the term $F_j(\tau-1)$. The normalisation term $N_t = P(X_t = x_t \mid x_0^{t-1} = x_0^{t-1})$ can be computed during the forward iteration using the same decomposition as for the forward algorithm.

The backward algorithm is used to compute the smoothed marginal

$$\begin{aligned} L_j(t) =& P(S_t = j \mid X_0^{\tau-1} = x_0^{\tau-1}) \\ =& P(S_{t+1} \neq j, S_t = j \mid x_0^{\tau-1}) + P(S_{t+1} = j \mid x_0^{\tau-1}) - \\ & P(S_{t+1} = j, S_t \neq j \mid x_0^{\tau-1}) \\ =& L1_j(t) + L_j(t+1) - P(S_{t+1} = j, S_t \neq j \mid x_0^{\tau-1}) \end{aligned}$$

Thus we need to compute $L1_j(t)$ and $P(S_{t+1} = j, S_t \neq j \mid x_0^{\tau-1})$. For $t = \tau-2, \ldots, 0, j = 0, \ldots, J-1$, $L1_j(t)$ can be computed iteratively as

$$\begin{aligned} L1_j(t) =& P(S_{t+1} \neq j, S_t = j \mid x_0^{\tau-1}) \\ =& \sum_{k \neq j} [\sum_{u=1}^{\tau-t-2} P(S_{t+u+1} \neq k, S_{t+1}^{t+u} = k, S_t = j \mid x_0^{\tau-1}) \\ & + P(S_{t+1}^{\tau-1} = k, S_t = j \mid x_0^{\tau-1})] \end{aligned} \tag{12}$$

To re-estimate the state duration distribution functions, we also need to compute

$$\begin{aligned} \eta_{ju} =& \sum_{t=0}^{\tau-2} P(S_{t+u+1} \neq j, S_{t+1}^{t+u} = j, S_t \neq j \mid X_0^{\tau-1} = x_0^{\tau-1}) \\ & + P(S_u \neq j, S_0^{u-1} = j \mid X_0^{\tau-1} = x_0^{\tau-1}) \end{aligned}$$

Now we get the complete recursion formula to compute the backward probability. In fact, the recursion formula can be simplified by using a quantity

$$G_j(t+1, u) = \frac{L1_j(t+u)}{F_j(t+u)} \prod_{v=1}^{u} \frac{e_j(x_{t+v})}{N_{t+v}} d_j(u), \quad u = 1, \ldots, \tau-2-t \tag{14}$$

2.3 State duration distribution

If the state duration probability distributions are estimated in a non-parametric manner, the re-estimation formula for state j is as follows

$$d_j^{(i+1)}(u) = \frac{\eta_{ju}}{\sum_v \eta_{jv}} = \frac{\eta_{ju}}{\sum_{t=0}^{\tau-2} L1_j(t) + L_j(\tau-1)} \tag{15}$$

The state duration distributions may also be modelled by parametric probabilistic distributions. As suggested in Guedon (2003), the quantity η_{ju} can be considered as a pseudo-sample generated from a parametric state duration distribution for state j. Therefore, we can use classic point estimation procedures from η_{ju} to estimate the parameters of $d_j(u)$. For example, for Poissoion state duration distribution $p(u \mid \lambda) = \frac{\lambda^{(u-1)} e^{-\lambda}}{(u-1)!}$, the parameter λ_j for the jth state at the ith iteration can be estimated by

$$\lambda_j^{(i+1)} = \frac{\sum_{u=1}^{M} \eta_{ju} * (u-1)}{\sum_{u=1}^{M} \eta_{ju}} \tag{16}$$

We can get this formula by plugging $d_j(u)$ into equation (1), and then take the partial derivative with respect to λ_j and set to zero. For geometric duration distribution $p(u \mid \theta) = (1-\theta)^{u-1}\theta$, the parameter θ_j can be estimated as follows

$$\theta_j^{(i+1)} = \frac{\sum_{u=1}^{M} \eta_{ju}}{\sum_{u=1}^{M} u\eta_{ju}} \tag{17}$$

Both Poisson and geometric distributions are special cases of the negative binomial distribution $\mathrm{NB}(u \mid r, \theta) = \binom{u-2+r}{u-1} \theta^r (1-\theta)^{u-1}$. For this distribution, r can be estimated by minimising the negative log likelihood function

$$\begin{aligned} \theta_j &= \frac{r \sum_u \eta_{ju}}{\sum_u \eta_{ju}(r+u-1)} \\ r^{(i+1)} &= \arg\min_r -\sum_{u=1}^{D} \eta_{ju} [\log\Gamma(u-1+r) - \log\Gamma(r) \\ &\qquad + r\log\theta_j + (u-1)\log(1-\theta_j)] \end{aligned} \tag{18}$$

After estimating $r^{(i+1)}$, the parameter $\theta_j^{(i+1)}$ can be estimated by

$$\theta_j^{(i+1)} = \frac{r^{(i+1)} \sum_u \eta_{ju}}{\sum_u \eta_{ju}(r^{(i+1)} + u - 1)} \tag{19}$$

2.4 The initial distribution and state translation probabilities

The initial probabilities and the translation probabilities can be easily obtained from the corresponding Q functions by Lagrange multiplies. The final formulas are as follows:

$$\pi_j^{(i+1)} = P(S_0 = j \mid x_0^{\tau-1}, \theta^{(i)}) = L_j(0) \tag{20}$$

For the translation probability,

$$\begin{aligned} A_{ij}^{(i+1)} &= \frac{\sum_{t=0}^{\tau-2} P(S_{t+1}=j, S_t=i \mid x_0^{\tau-1}, \theta^{(i)})}{\sum_{t=0}^{\tau-2} P(S_{t+1}\neq i, S_t=i \mid x_0^{\tau-1}, \theta^{(i)})} \\ &= \frac{\sum_{t=0}^{\tau-2} G_j(t+1) A_{ij} F_i(t)}{\sum_{t=0}^{\tau-2} L1_i(t)} \end{aligned} \tag{21}$$

2.5 The Viterbi algorithm

Based on the smooth probability $L_j(t)$, we can find the most probable sequence of states. However, the marginally most likely states may not constitute a legitimate path. To find the most probale path $s_0^{*\tau-1}$ given a sequence $x_0^{\tau-1}$, i.e.,

$$s_0^{\star\tau-1} = \arg\max_{s_0^{\tau-1}} P(s_0^{\tau-1}, x_0^{\tau-1})$$

we can use the Viterbi algorithm (Bishop, 2006). As for HMMs, the Viterbi algorithm is the same as the Forward algorithm except that the 'sum' operator is replaced by the 'max' operator.

3 Results

In this section, we will use the proposed robust HSMM to model aCGH data. aCGH is a high-throughput technique for identifying DNA copy number variations which is central to understanding some genetic diseases, such as Cancers (Mitrofanova and Mishra, 2010). Experiments are conducted on two data sets. The first one is the bacterial artificial chromosome array hybridisations of coriell cell lines data set (Snijders et al., 2001), which is considered as the golden standard data set. The second one is the Glioblastoma Multiforme (GBM) data set (Bredel et al., 2005), which has been used to evaluate 11 aCGH copy number variation detection algorithms (Lai et al., 2005). This data set has also been used in Guha et al. (2008) and Lai et al. (2008) to assess their methods. These two data sets are representative because the first one has high Signal-to-Noise ratio (S/N) while the second one has low S/N.

3.1 Coriell cell lines

This data set is derived from 15 primary breast tumors, and the karyotypes are available from the supplementary Table 1 of Snijders et al. (2001). Therefore, we can evaluate our method by comparing the Viterbi paths with the cytogenetically mapped alterations. Based on the annotations, our robust HSMM has 3 states, and they correspond to copy-loss, copy-neutral and copy-gain, respectively. We train the robust HSMM on the whole data set, which means the parameters are estimated using pooled data across all the chromosomes and all the cell lines. The Viterbi algorithm is used to find the most likely path for a given sequence of a chromosome.

The Coriell data set has relatively high S/N. It may be sufficient to use one Student component. We first train an HMM with an SMM emission distribution function. Table 1 shows the results. There are 15 short segment False Positives (FP). The reason why we have these relatively large number of FPs is that the state duration distributions may be complex, especially when we pool data. The geometric state duration distribution functions of HMMs have large values at the beginning. Therefore, they favour short segments. For HSMMs, the state duration distributions for the copy-loss and copy-gain state are modelled by Poisson distributions, and for the copy-neutral state, the distribution is modelled by a negative binomial distribution. Figure 2 shows the estimated state duration distribution functions of the HMM+SMM and HSMM+SMM model. (The HMM+GMM does not show good performance, so the results are not given).

Figure 2 The estimated state duration distributions $d_j(u)$ of model: (a) HMM+SMM and (b) HSMM+SMM (see online version for colours)

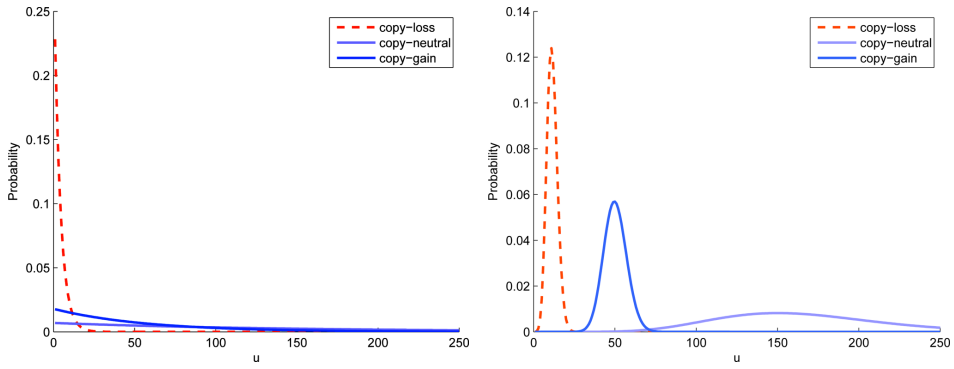


Table 1 Results from applying different models to the Coriell data set. Here FP is copy-neutral misclassified as copy-gain or copy-loss, and FN is copy-gain or copy-loss misclassified as copy-neutral

Model	*FP*	*FN*
HMM + SMM (1)[a]	15	1
HSMM + GMM (2)	1	1
HSMM + SMM (1)	1	1
CBS[b] (Olshen et al., 2004)	1	3
CGH_{SEG}[b] (Picard et al., 2005)	8	1
$CGH_{TRIMMER}$[b] (Tsourakakis et al., 2010)	1	1

[a]SMM(1) means an SMM with one student component and GMM(2) means a GMM with 2 Gaussian components.

From Tsourakakis et al. (2010), Table 3.

Table 1 also shows the results by using the HSMM + SMM and HSMM + GMM model. Although they give the same results, the SMM uses only one Student component (when two Students components are used, the model performs equally well), while the GMM uses two Gaussian components. One component models the expected $\log_2$ ratio for each

state j, and the other component models the $\log_2$ ratio generated by outliers of the state. This is the robust HMM, introduced in Shah et al. (2006) to model aCGH data. Based on the experiments, we suggest using SMMs because they are more stable than the GMMs. One example is that the HMM + GMM model performs poorly on this data set (results not shown here) while the HMM + SMM model still works reasonable. In addition, the Bayesian HMM (Guha et al., 2008) prediction results have many false positives.

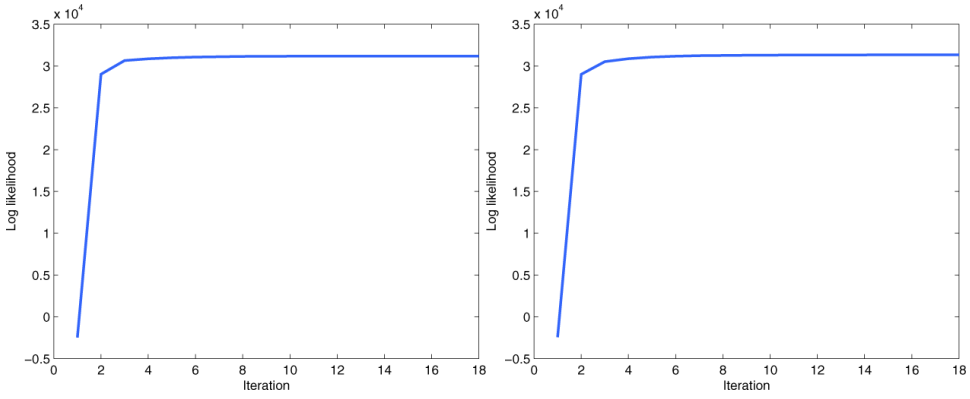
We also compare the proposed model with the CBS algorithm (Olshen et al., 2004), CGH_{SEG} (Picard et al., 2005) and a recently proposed algorithm $\text{CGH}_{\text{TRIMMER}}$ (Tsourakakis et al., 2010). The results are also given in Table 1. Our robust HSMM model performs equally well as the $\text{CGH}_{\text{TRIMMER}}$ algorithm. It is interesting to notice that the $\text{CGH}_{\text{TRIMMER}}$ algorithm and our robust HSMM make the same false positive along chromosome 1, cell line GM03576 (based on our experiments, not the results given in the paper (Tsourakakis et al., 2010). This is a one-point segment. The false negative is the copy-loss in cell line GM07081, chromosome 7. For this chromosome, there is no evidence of alteration in the data. No methods can detect the copy-loss.

Although the $\text{CGH}_{\text{TRIMMER}}$ algorithm and our algorithm performs equally well on this data set, we still prefer our method because

- the parameters of our model have clear biology meanings
- our model can easily take account of information across the whole chromosomes and all the cell lines
- the $\text{CGH}_{\text{TRIMMER}}$ algorithm does not assume a probability distribution of the observations. It is not a robust method, and may perform poorly on noisy data (see the next section).

Figure 3 shows the log likelihoods at each iteration of the EM algorithms for both robust HMM and HSMM. From the figures we can see the EM algorithms converge rapidly. Table 2 shows the robust HSMM obtained after the EM algorithms converge.

Figure 3 Plots of the log likelihood's as a function of iterations for (a) HMM+SMM and (b) HSMM + SMM. The two plots are quite similar and the EM algorithms converge rapidly (see online version for colours)



(a) (b)

Table 2 The initial parameters and the estimated parameters from the EM algorithm

	Initial	Estimated
Transition A_{ij}	$\begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.5 & 0.0 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0 & \approx 1 & \approx 0 \\ 0.3984 & 0 & 0.6016 \\ \approx 0 & \approx 1 & 0 \end{bmatrix}$
Emission	$\begin{bmatrix} \mu \mid -1 & 0 & 0.58 \\ \Sigma \mid 0.1 & 0.1 & 0.1 \\ \nu \mid 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -0.7533 & -0.0017 & 0.4865 \\ 0.0129 & 0.005 & 0.0085 \\ 3.9462 & 5.5776 & 5.8527 \end{bmatrix}$
Prior π_j	[0.1, 0.8, 0.1]	[0.003, 0.9636, 0.0333]
Duration	$\begin{bmatrix} Poiss & NB & Poiss \\ 5 & \begin{matrix} 2 \\ 0.008 \end{matrix} & 40 \end{bmatrix}$	$\begin{bmatrix} 10.4468 & \begin{matrix} 11.2768 \\ 0.0642 \end{matrix} & 49.1860 \end{bmatrix}$

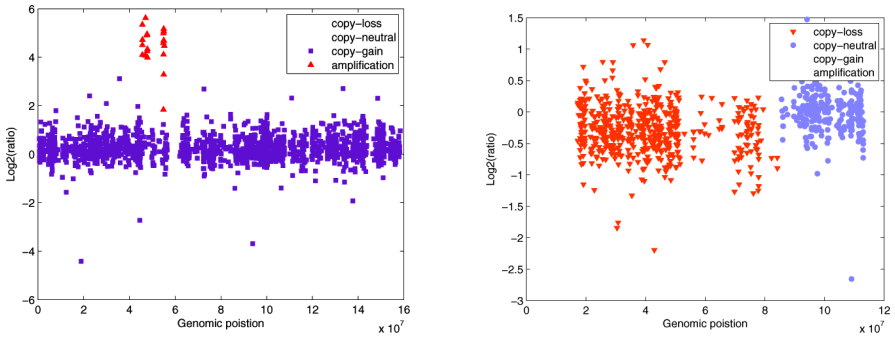
3.2 Glioblastoma Multiforme (GBM) data

This data set contains measurements from 26 primary Glioblastoma Multiforme tumors. Compared with the Coriell data set, the GBM data set is noisy. It contains large but low amplitude regions of copy-losses and copy-gains, and small but high amplitude regions of amplifications (Lai et al., 2005). Therefore, we use a 4-state HSMM as in Guha et al. (2008). These states represent copy-loss, copy-neutral, copy-gain and amplification, respectively.

Previous reports only gave the results of two chromosomes with typical alternations: GBM29 chromosome 7 and GBM31 chromosome 13. Follow an identical evaluation procedure, Figure 4 shows the results. Our robust HSMM correctly detects the three high amplitude regions of amplification in GBM29 chromosome 7, and the low amplitude region of copy-loss in GBM31, chromosome 13. Note for chromosome 7 of GBM29, our algorithm classifies the remaining parts as copy-gain. Based on the supplementary Table 2 of Bredel et al. (2005), they already detect the gain in copy number at p12.3, p13, p14.1, p15.1, p15.3, p21.1, p21.2, q21.1, q21.3, q21.11-13, q22.3, q31.1-2, q31.33, q32.3, q36.1.

Because of the low S/N, the $\text{CGH}_{\text{TRIMMER}}$ algorithm doesn't perform well on this data set. In fact, the above 2 chromosomes are relatively easy to segment. For other chromosomes, the Bayesian HMM (Guha et al., 2008) also does not perform well enough partially because it is not a robust method, and the parameters are not estimated using pooling.

Figure 4 The estimated state for each clone of (a) GBM29, chromosome 7 and (b) GBM31, chromosome 13 (see online version for colours)



4 Conclusions and future work

In this paper, we have developed a robust HSMM, and applied the model to analyse aCGH data. Compared with HMMs, the robust HSMM is more suitable to analyse sequences with succession of homogenous zone by explicitly modelling the state duration distributions. The SMM emission distributions have large tails and provide protection against outliers. Therefore, the model is less likely to be disturbed by small segment outliers. Experiments show the superior performance of the proposed model on data with high S/N or low S/N.

Because HMMs are special cases of HSMMs when the state duration distributions are geometric distributions, we check the correctness of our code by comparing our program's outputs with those of the HMM implemented in the PMTK package (http://code.google.eom/p/pmtk3/). The two programs give exactly the same results when geometric state duration distributions are used. The software, written in Matlab and C, is available from (https://www.cs.ubc.ca/~jiaruid/software/rhsmm), so readers can reproduce all the results.

As the time complexity of the inference and learning algorithms for HSMMs is higher than for HMMs, one interesting direction for future work is to design approximate algorithms such as approximate Viterbi algorithms. Also, it would be interesting to consider Markov Chain Monte Carlo sampling algorithms. As for the applications of our model, we want to apply the model to high-resolution genotyping array data.

Acknowledgements

The authors are grateful to Anne Condon from the Department of Computer Science, University of British Columbia for many helpful suggestions.

References

Bilmes, J. (1997) '*A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*, Technical Report TR-97-021, ICSI.

Bishop, C.M. (2006) *Pattern Recognition and Machine Learning,* Springer, New York.

Bredel, M., Bredel, C., Juric, D., Harsh, G., Vogel, H., Recht, L. and Sikic, B. (2005) 'High-resolution genome-wide mapping of genetic alterations in human glial brain tumors', *Cancer Research,* Vol. 65, No. 10, pp.4088–4096.

Chatzis, S., Kosmopoulos, D. and Varvarigou, T. (2009) 'Robust sequential data modelling using an outlier tolerant hidden markov model', *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 31, No. 9, pp.1657–1669.

Cheng, C., Sha, F. and Saul, L. (2010) 'Online learning and acoustic feature adaptation in large-margin hidden Markov models', *Selected Topics in Signal Processing, IEEE Journal of,* Vol. 4, No. 6, pp.926–942.

Guedon, Y. (2003) 'Estimating hidden semi-Markov chains from discrete sequences', *Journal of Computational and Graphical Statistics,* Vol. 12, No. 3, pp.604–639.

Guha, S., Li, Y. and Neuberg, D. (2008) 'Bayesian hidden Markov modeling of array CGH data', *Journal of the American Statistical Association*, Vol. 103, No. 482, pp.485–497.

Lai, T., Xing, H. and Zhang, N. (2008) 'Stochastic segmentation models for array-based comparative genomic hybridization data analysis', *Biostatistics*, Vol. 9, No. 2, pp.290–307.

Lai, W., Johnson, M., Kucherlapati, R. and Park, P. (2005) 'Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data', *Bioinformatics*, Vol. 21, No. 19, pp.3763–3770.

Mitrofanova, A. and Mishra, B. (2010) 'On a novel coalescent model for genome-wide evolution of copy number variations', *International Journal of Data Mining and Bioinformatics*, Vol. 4, No. 3, pp.300–315.

Murphy, K.P. (In preparation) *Machine Learning, A Probabilistic Perspective*, MIT Press, Cambridge, MA.

Olshen, A., Venkatraman, E., Lucito, R. and Wigler, M. (2004) 'Circular binary segmentation for the analysis of array-based DNA copy number data', *Biostatistics*, Vol. 5, No. 4, pp.557–572.

Peel, D. and McLachlan, G. (2000) 'Robust mixture modelling using the t distribution', *Statistics and Computing*, Vol. 10, No. 4, pp.339–348.

Picard, F., Robin, S., Lavielle, M., Vaisse, C. and Daudin, J. (2005) 'A statistical approach for array CGH data analysis', *BMC Bioinformatics*, Vol. 6, No. 27, URL: http://www.biomedcentral.com/1471-2105/6/27 (Accessed 8 February 2012)

Shah, S., Xuan, X., DeLeeuw, R., Khojasteh, M., Lam, W., Ng, R. and Murphy, K. (2006) 'Integrating copy number polymorphisms into array CGH analysis using a robust HMM', *Bioinformatics*, Vol. 22, No. 14, pp.e431–e439.

Snijders, A., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, J., Hamilton, G., Hindle, A., Huey, B., Kimura, K., Law, S., Myambo, K., Palmer, J., Ylstra, B., Yue, J., Gray, J., Jain, A., Pinkel, D. and Albertson, D. (2001) 'Assembly of microarrays for genome-wide measurement of DNA copy number', *Nature Genetics*, Vol. 29, No. 3, pp.263–264.

Tsourakakis, C., Tolliver, D., Tsiarli, M., Shackney, S. and Schwartz, R. (2010) 'CGHTRIMMER: Discretizing noisy Array CGH Data', Arxiv preprint arxiv: 1002.4438, URL: http://arxiv.org/abs/1002.4438 (accessed 8 February 2012).

Yu, S. (2010) 'Hidden semi-Markov models', *Artificial Intelligence*, Vol. 174, No. 2, pp.215–243.