Multiresolution Mixture Modeling using Merging of Mixture Components

Prem Raj Adhikari PREM.ADHIKARI@AALTO.FI
Jaakko Hollmén JAAKKO.HOLLMEN@AALTO.FI
Helsinki Institute for Information Technology and Department of Information and Computer Science
Aalto University School of Science, PO Box 15400, FI-00076 Aalto, Espoo, Finland

Editor: Steven C.H. Hoi and Wray Buntine

Abstract

Observing natural phenomena at several levels of detail results in multiresolution data. Extending models and algorithms to cope with multiresolution data is a prerequisite for wide-spread exploitation of the data represented in the multiple resolutions. Mixture models are widely used probabilistic models, however, the mixture models in their standard form can be used to analyze the data represented in a single resolution. In this paper, we propose a multiresolution mixture model based on merging of the mixture components across models represented in different resolutions. Result of such an analysis scenario is to have multiple mixture models, one mixture model for each resolution of data. Our proposed solution is based on the idea on the interaction between mixture models. More specifically, we repeatedly merge component distributions of mixture models across different resolutions. We experiment our proposed algorithm on the two real-world chromosomal aberration datasets represented in two different resolutions. Results show an improvement on the compared multiresolution settings.

Keywords: Multiresolution, Mixture Models, KL Divergence, 0-1 data

1. Introduction

Multiresolution data arise when an object or a phenomenon is described at several levels of detail. Multiresolution data are prevalent in several application domains such as image processing, hydrology, telecommunications, time series analysis, and astronomy (Willsky, 2002). The notion of multiresolution models is also related with the notion of multi-scale modelling (Cristini and Lowengrub, 2010; Ferreira and Lee, 2007) and wavelets (Mallat, 1989), thus widening the perspective of research on the multiresolution analysis.

Finite mixture models, or shortly mixture models, are probabilistic models widely used in the several analysis tasks such as clustering, density estimation, handling missing data, and modelling heterogeneity (Bishop, 2006; McLachlan and Peel, 2000). Mixture models are one of the most popular probabilistic modelling techniques due to their relative simplicity and flexibility to model more complex distributions based on a superposition of simple, parametric component distributions. In their standard form, however, the mixture models can be used to analyze the data in a single resolution.

A straightforward extension to handle data in the multiple resolutions is to model the data represented in different resolutions separately and to compare or combine the obtained results. Another option is to ignore one source of data and model only one of the available data sets, which is represented in a single resolution. This effort results in less data and less representative data. The improvement on this method is to transform the data in different resolutions to a single resolution, integrate the data sets, and then apply the mixture models on the integrated data in the same resolution. This improves the performance over single analysis on the data in the different resolutions separately which has been shown in our previous work (Adhikari and Hollmén, 2010a,b).

All the previously mentioned solutions generate models in a single resolution. A more natural setting would be to have a separate model for each of the resolution, each of the models reflecting the properties of the data sets jointly. Our problem scenario is such that there are two or more datasets describing the same domain and which are expected to have the same distribution, but they have a different data dimensionality, or resolution. We learn the mixture models for each resolution so that we also model the interaction across different resolutions. Authors have tried to use the mixture models for the multiresolution data especially in the image processing domain (Wilson, 2000). However, the mixture of trees used in image compression and reconstruction in (Wilson, 2000) are not directly applicable in the other applications because the pyramid structure and the scale space in other applications are not as smooth as the one in the image processing. Furthermore, the data (features) in biology are often irregular thus necessitating a specialized approach to analyze the multiresolution data. In this paper, we propose a multiresolution mixture model based on the idea of merging of the mixture components across the different resolutions.

The concept of splitting and merging of the mixture components keeping their number fixed is used in (Ueda et al., 2000) and (Zhang et al., 2003) to ameliorate the problem of the local minima in the EM algorithm. Similarly, the authors in (Li and Li, 2009) and (Adhikari and Hollmén, 2012) use the split and merge strategy combined with a model selection criterion such as the Minimum Description Length (MDL) and the cross-validation (CV) to determine the optimal number of the mixture components in a mixture model (i.e. for model selection) varying the number of mixture components to search for the optimal number of the mixture components. The authors used the components within the same mixture model and only the two components are merged at any instant. In our proposed multiresolution mixture model, in contrast, we often merge more than the two mixture components from more than the two different mixture models. Similarly, in all of these studies, the authors do not consider the mixture models for the data in the multiple resolutions.

We train the mixture model separately in the different resolutions and merge the mixture components in the different resolutions thereby producing the mixture models in the multiple resolutions. We use the data driven fast approximation of the KL divergence to compute the similarity between the mixture components.

Our proposed algorithm of the multiresolution modelling is similar to clustering aggregation (Gionis et al., 2007), which for a given many clusterings generates a single clustering that agrees as much as possible with the initial, input clusterings. Here, we can view the mixture model in different resolutions as the different input clusterings and the multiresolution mixture model as the model that aggregates (agrees as much as possible) the information on the mixture models in the different resolutions. Our proposed algorithm produces the multiple mixture models in the different resolutions whereas the clustering aggregation produces only a single clustering result. Furthermore, clustering aggregation works in data-space where as our proposed algorithm works in model-space.

A single clustering algorithm can not be the best clustering algorithm for every situation and every dataset. Similar to the clustering aggregation, several clustering ensemble algorithms have been proposed with an aim to combine the different partitions obtained by the different clustering algorithms into a single clustering solution (Ghaemi et al., 2009; Vega-Pons and Ruiz-Shulcloper, 2011). Clustering ensemble improves the clustering solution with respect to the robustness, novelty, stability and confidence estimation, and parallelization and scalability (Topchy et al., 2004; Ghaemi et al., 2009). However, these clustering ensemble methods use the consensus functions such as relabeling, voting, mutual information, and co-association which are not directly usable neither in the mixture models nor in the multiple resolutions.

Topchy et al. (2004) used the mixture models for clustering ensembles but the results of multiple clustering methods are used as input features to the finite mixture models. The limitation of the method is that it is suitable for the data in a single resolution and the final models are also available only in a single resolution. Furthermore, the mixture models used in (Topchy et al., 2004) are not valid for patterns in the original space. However, we can reap the benefits of generative property of the mixture models only if the model is valid for the data in the original space. Additionally, the information while modelling the multiresolution phenomenon is best preserved while modelling in multiple resolutions simultaneously. Therefore, instead of modelling each resolution separately, we absorb the information contained in different resolutions in a single model and generate the models in multiple resolutions. We experiment the proposed algorithm on the chromosomal aberrations data in the multiple resolutions.

The rest of the paper is organized as follows. Section 2 briefly reviews the mixture models of the multivariate Bernoulli distributions. Section 3 discusses and derives the KL Divergence to compare the mixture components in the different mixture models in the multiple resolutions. Section 4 discusses the process of transforming the parameters of the mixture models of different dimensionality across different resolutions. Section 5 presents our proposed multiresolution mixture modelling algorithm. Section 6 contains the description of the experiments performed on the real-world dataset describing the chromosomal aberrations in two different resolutions. Section 7 summarizes the paper.

2. Mixture Models for 0-1 Data

Finite mixture models of multivariate Bernoulli distributions (Wolfe, 1970), composed as a sum of J component distributions are defined as

$$p(\boldsymbol{x}|\boldsymbol{\Theta}) = \sum_{j=1}^{\mathrm{J}} \pi_j \prod_{i=1}^{\mathrm{d}} \theta_{ji}^{x_i} (1-\theta_{ji})^{1-x_i}. \tag{1}$$

The data vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$ consists of d elements, and $x_i \in \{0, 1\}$. The mixture proportions π_j satisfy the properties $\pi_j \geq 0$, $\forall j = 1, \ldots J$ and $\sum_{j=1}^{\mathrm{J}} \pi_j = 1$. The component distributions are parametrized with the Bernoulli parameters θ_{ji} containing parameters for each component distribution j and for each data vector element i. We can collect the mixture coefficients to a vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J)$, and the parameters of the component distributions to a matrix $\boldsymbol{\Theta} = (\theta_{ji})$. The parameters of the mixture model of multivariate

Bernoulli distributions are now $\{J, \boldsymbol{\pi}, \boldsymbol{\Theta}\}$. Learning the mixture model requires estimation of the number of the mixture components, J, the mixing proportions, $\boldsymbol{\pi}$, and the parameters of the component distributions, $\boldsymbol{\Theta}$. If the number of mixture components J is assumed to be known, the EM algorithm (Dempster et al., 1977) can be used to learn the maximum likelihood estimates for the parameters of the mixture model.

Model selection in the context of mixture models refers to the problem of selecting an appropriate number of the component distributions. Several model selection algorithms have been proposed in the literature to learn the number of the mixture components in the mixture model (Smyth, 2000; Figueiredo and Jain, 2002). In our previous work, we have demonstrated the use of the model selection algorithms in the mixture models of the multivariate Bernoulli distributions (Tikka et al., 2007; Hollmén and Tikka, 2007; Adhikari and Hollmén, 2010a,b; Adhikari and Hollmén, 2012). In this paper, we are not concentrated on the problem of model selection but the proposed algorithm uses the trained models in the multiple resolutions and absorbs the information in multiple resolutions thereby generating models in the multiple resolutions.

3. Kullback-Leibler Divergence

Kullback-Leibler (KL) divergence is a non-symmetric measure of the difference between the two probability distributions (Kullback and Leibler, 1951; Kullback, 1959). Given two probability distributions P and Q, the KL divergence can be symmetrized by averaging the KL divergence from P to Q and from Q to P (Dagan et al., 1997). Mathematically, the symmetric KL divergence between the two probability distributions P and Q is given by:

$$\begin{aligned} \mathcal{D}_{KL}(P||Q) + \mathcal{D}_{KL}(Q||P) &= \sum_i P(i) log \frac{P(i)}{Q(i)} + \sum_i Q(i) log \frac{Q(i)}{P(i)} \\ &= \sum_i \left[\{P(i) - Q(i)\} log \frac{P(i)}{Q(i)} \right] \end{aligned} \tag{2}$$

where i indexes all the possible combinations of data elements.

The KL divergence between the two components in a mixture model to compare the two component distributions from a mixture model of the multivariate Bernoulli distributions has been derived in (Adhikari and Hollmén, 2012) as:

$$KL_{\alpha\beta} = \sum_{i \in X^*} \left\{ \prod_{k=1}^{d} \left(\alpha_k^{X^*_{ik}} (1-\alpha_k)^{(1-X^*_{ik})} \right) - \prod_{k=1}^{d} \left(\beta_k^{X^*_{ik}} (1-\beta_k)^{(1-X^*_{ik})} \right) \right\}. \tag{3}$$

Here, i indexes the unique samples denoted by X^* such that $X^* = \{x^* : x^* \in \underline{\boldsymbol{X}}\}$ where $\underline{\boldsymbol{X}}$ denotes the dataset. The two component distributions in a mixture model are denoted by α and β. Similarly, d denotes the dimensionality of data indexed by k. The Equation (3) constraints that both the component distributions should have the same dimensionality and should be indexed by the same dataset. We can extend this comparison to the multiresolution scenario under the simple and realistic assumption that the difference in the dimensionality contributes very less to the difference in the KL divergence as:

$$KL = \sum_{i\in X^*} \pi_\alpha \prod_{m=1}^{d} \left(\alpha_m^{X^*_{im}}(1-\alpha_m)^{(1-X^*_{im})}\right) - \sum_{i\prime\in Y^*} \pi_\beta \prod_{n=1}^{d'} \left(\beta_n^{Y^*_{i\prime n}}(1-\beta_n)^{(1-Y^*_{i\prime n})}\right) \quad (4)$$

Here, i and $i\prime$ indexes the unique samples in the two different datasets in two resolutions denoted by $\underline{\overline{\boldsymbol{X}}}$ and $\underline{\overline{\boldsymbol{Y}}}$ such that $X^* = \{x^* : x^* \in \underline{\overline{\boldsymbol{X}}}\}$ and $Y^* = \{y^* : y^* \in \underline{\overline{\boldsymbol{Y}}}\}$ are the set of all the unique data samples present in each dataset, respectively. Additionally, m and n indexes the dimensionality of datasets in the coarse and the fine resolution denoted by d and d', respectively. Here, α and β denote the component distributions in the two different mixture models in two different resolutions. Since Equation (4) approximates the symmetric KL divergence, the two terms in the equation can be interchanged. Furthermore, the Equation (4) is also suitable for cases when the number of data samples in the two different resolutions are different. In addition to the approximations, we weigh the KL divergence with their respective mixing proportions denoted by π_α and π_β in the Equation (4). When the KL divergence is weighted with the mixing proportions, it also considers the similarity of the mixing proportions which adds more suitability to comparing the component distributions from the different mixture models. Additionally, it is more desirable to merge the mixture components having the higher mixing proportions or having the lower mixing proportions as the mixing proportions also carries the information about similarity of the two mixture components in the context of the two different mixture models.

4. Sampling of Model Parameters

Merging the mixture components in the different models in the different resolutions is not straightforward because of the difference in the number of parameters (i.e. dimensionality d of the model parameters, $\boldsymbol{\Theta}$) of the component distributions. Therefore, we upsample the model parameters of the component distributions of the mixture models in the coarse resolution and downsample the parameters of the component distributions in the fine resolution to ensure that the dimensionality of the model parameters are the same. The concept of upsampling and downsampling is similar to that in the multiresolution data proposed in (Adhikari and Hollmén, 2010b) so that data in the different resolutions could be integrated. However, this paper proposes the upsampling and downsampling of the model parameters which allows seamless and simultaneous modelling of the multiresolution data. The model parameters are probabilities, not the 0-1 data as in (Adhikari and Hollmén, 2010b), therefore the upsampling and downsampling methods differs from the ones proposed in (Adhikari and Hollmén, 2010b). Furthermore, the sampling is performed in the model-space and not data-space thus providing simultaneous and seamless modelling of the multiresolution data.

4.1. Upsampling the model parameters

Upsampling transforms the model parameters of the component distributions from the coarse resolution to the fine resolution. In this case, one model parameter in the coarse resolution should produce multiple parameters in the fine resolution. We upsample the single model parameter by re-sampling the number of chromosomal regions required in

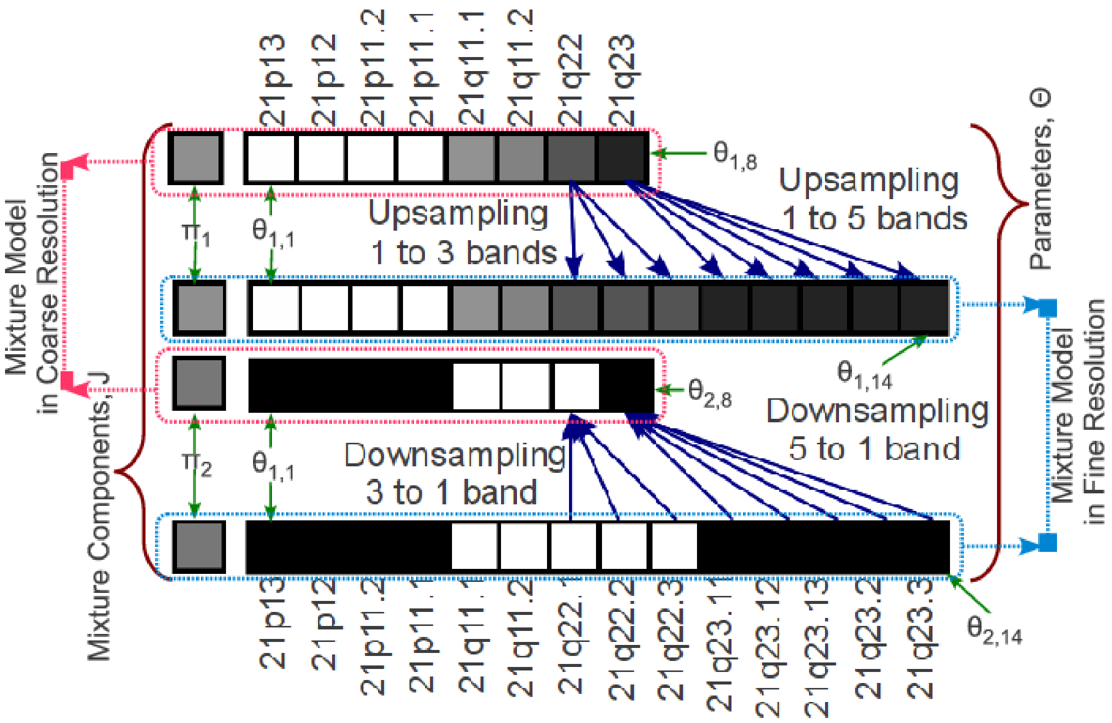

Figure 1: Illustration of the upsampling and the downsampling procedure for the model parameters in the two mixture models in the two resolutions. This is an example case in chromosome 21. The model parameters denote the regions of chromosome 21 and some of the chromosomal regions are unchanged across different resolutions as shown in (Shaffer and Tommerup, 2005). These unchanged chromosomal regions are not altered during sampling. However, other regions are upsampled from the coarse resolution and downsampled from the fine resolution according to the division of the chromosomal regions across different resolutions.

the fine resolution from a Normal distribution where the value of the model parameter in the coarse resolution is the mean and standard deviation is a small value (0.01 in our experiments). Since the model parameters are probabilities, we restrict the values of the model parameters θ between 0 and 1 $\{0 \leq \theta \leq 1\ \}$ by replacing the values that violate this constraint with the value of the model parameter in coarse resolution. Nevertheless, such deviations are rare because the value of standard deviation is small.

4.2. Downsampling the model parameters

Downsampling transforms the model parameters of the component distributions from the fine resolution to the coarse resolution combining the multiple parameters in the fine resolution to form a single parameter in the coarse resolution. We estimate the mean and standard deviation from the model parameters of the component distributions that are to be combined to downsample the model parameters. We then re-sample one model parameter from a Normal distribution with estimated mean and the standard deviation. Similar to the upsampling, any value deviating from the probability range of $\{0 \leq \theta \leq 1\ \}$ is replaced with the mean of the model parameters in the fine resolution.

Figure 1 shows the upsampling and downsampling of the model parameters between the two different mixture models having two mixture components each. One detached block in the left in all the four rows visualizes the mixture components. The eight and fourteen adjoining blocks to the right shows the parameters of the component distributions. Darker color represents higher value for the mixture components and the model parameters while the lighter color denotes the smaller value for the mixture components and the model parameters. The mixing proportions are not changed while downsampling and upsampling. The solid arrows between the components across the different model parameters denotes the upsampling and downsampling procedures. The downward pointing arrows represent upsampling while the upward pointing arrows represent downsampling. The dotted arrows depict the two mixture models in two different resolutions.

4.3. Merging of Mixture Components

We select the components to be merged using the minimum weight bipartite matching (West, 1996) from the calculated symmetric KL divergence between the different components in the different mixture models. The updates are made in all the component distributions in all the mixture models initially by averaging all mixing proportions to be merged from the different mixture models as in the Equation (5) and subsequently to all the mixture components in a single mixture model during the normalization as in the Equation (6).

$$\pi_{merged} \;=\; \frac{\pi_{klmin,1} \;+\; \pi_{klmin,2} + \ldots + \;\pi_{klmin,n}}{n} \tag{5}$$

Equation (5) averages the selected mixing proportions in the different mixture models. Here, $\pi_{klmin,1}$, $\pi_{klmin,2}$, $\ldots$, $\pi_{klmin,n}$ indexes the mixture components having the minimum KL divergence merged together to form π_{merged} in the merged model. The update by the Equation (5) could violate the constraints in the mixture model such as the convex combination and the sum of probabilities as discussed in Section (2). Hence, the mixing proportions in each mixture model are finally normalized according to the Equation (6).

$$\pi_j = \frac{\pi_j}{\sum_{j=1}^{J} \pi_j} \tag{6}$$

where $j = 1 \ldots J$ indexes all the components in the mixture model.

$$\Theta_{merged} \;=\; \frac{\pi_{klmin,1} \;\times\; \Theta_{klmin,1} \;+\; \pi_{klmin,2} \;\times\; \Theta_{klmin,2} + \ldots + \;\pi_{klmin,n} \;\times\; \Theta_{klmin,n}}{\pi_{klmin,n} \;+ \pi_{klmin,2} + \ldots + \pi_{klmin,n}} \tag{7}$$

The parameters can be merged according to the weight of the component distributions as given by the Equation 7. However, since the dimensionality of parameters of the component distributions are different, we upsample the parameters in the coarse resolution and downsample the parameters in the fine resolution. Secondly, we separately merge the mixture components in coarse resolution and the fine resolution using the Equation (7) producing merged model in each resolution.

5. Multiresolution Mixture Modelling Algorithm

Algorithm 1 provides the listings of the proposed algorithm to learn the multiresolution mixture model. The algorithm presents a simplified case of the multiresolution modelling which consists of the data in the two resolutions. The algorithm is scalable and expandable to N resolutions requiring $J(N-1)$ bipartite matching where J is the number of components. We do not need more than one comparison for any mixture model as we can move forward after selecting the similarity between the components in the first two mixture models. Given that a component a in the mixture model 1 is similar to a component b in the mixture model 2. If component c in mixture model 3 is similar to the component b in mixture model 2, then we can infer that the component a in the mixture model 1 is similar to the component c in the mixture model 3 without comparison. However, the parameters of the mixture model must be resampled (upsampled or downsampled) in all the different available resolutions.

Algorithm 1 Multiresolution Modelling using Merging of Mixture Components

```
Input: Two Datasets $\mathcal{D}_c$, $\mathcal{D}_f$, two mixture models $M_c$ and $M_f$ in coarse and fine resolution
    and a Threshold $\mathcal{T}_G$ for difference in KL divergence.
Output: Mixture Models $mm_c$ and $mm_f$ in coarse and fine resolution respectively that
    incorporates multiresolution data
klprev, indx ← 0
$\mathcal{T} \leftarrow \underset{k,l}{\text{argmax}} \ \mathcal{D}\{p(x \in \mathcal{D}_1; \Pi_k, \Theta_k); p(x \in \mathcal{D}_2; \Pi_l, \Theta_l)\}$
while $\mathcal{T}_G \leq \mathcal{T}$ do
  $merge_c$ , $merge_f \leftarrow \emptyset$
  for $i$ to min(Number of Components in $m_c$ ($\mathcal{J}_c$) and $m_f$ ($\mathcal{J}_f$) ) do
    $(k^*, l^*) \leftarrow \underset{k,l}{\text{argmin}} \ \mathcal{D}\{p(x \in \mathcal{D}_1; \Pi_k, \Theta_k); p(x \in \mathcal{D}_2; \Pi_l, \Theta_l)\}$
    where $k \in (1 \ldots \mathcal{J}_c)$, $l \in (1 \ldots \mathcal{J}_f)$ $k \notin merged_c$ , and $l \notin merged_f$
    $merge_c$, $merge_f$ ← insert $k^*$, $l^*$
    $m_{c2f}$, $m_{f2c}$ ← upsample($m_c$), downsample($m_f$)
    $mm_c$, $mm_f$ ← merge $\pi_{k^*}$ and $\pi_{l^*}$ in $m_{c2f}$ and $m_c$, and in $m_{f2c}$ and $m_f$
    $indx = indx + 1$
  end for
  if $mod(indx, 1000) == 0$ then
    $mm_c$, $mm_f$ ← Trained model on $\mathcal{D}_c$, $\mathcal{D}_f$ initialized using $mm_c$, $mm_f$
  end if
  $\mathcal{T} \leftarrow\mid klprev - \underset{k,l}{\text{argmin}} \mid$
  $klprev \leftarrow \underset{k,l}{\text{argmin}}$
end while
$mm_c$, $mm_f$ ← Trained model on $\mathcal{D}_c$, $\mathcal{D}_f$ initialized using $mm_c$, $mm_f$
return $mm_c$ and $mm_f$
```

The input to the algorithm is the two datasets and the two trained mixture models in the coarse and the fine resolution and a threshold to stop the iterations for minimizing the KL divergence. First, we calculate the symmetric KL Divergence between the different components of the mixture models in the different resolutions. Secondly, we upsample and downsample the model parameters so that they can be merged. We then match the components in the two different models using the minimum weight bipartite matching (West, 1996) and merge the components having the minimum KL Divergence. We retrain the mixture model via the EM algorithm initialized using the merged mixture model. We finally calculate the difference in the KL divergence between the two iterations. If the difference is less than the threshold, the algorithm ends by retraining the mixture models whereas if the change is not less than the threshold, we move on to the next iteration to minimize the KL divergence.

6. Experiments and Results

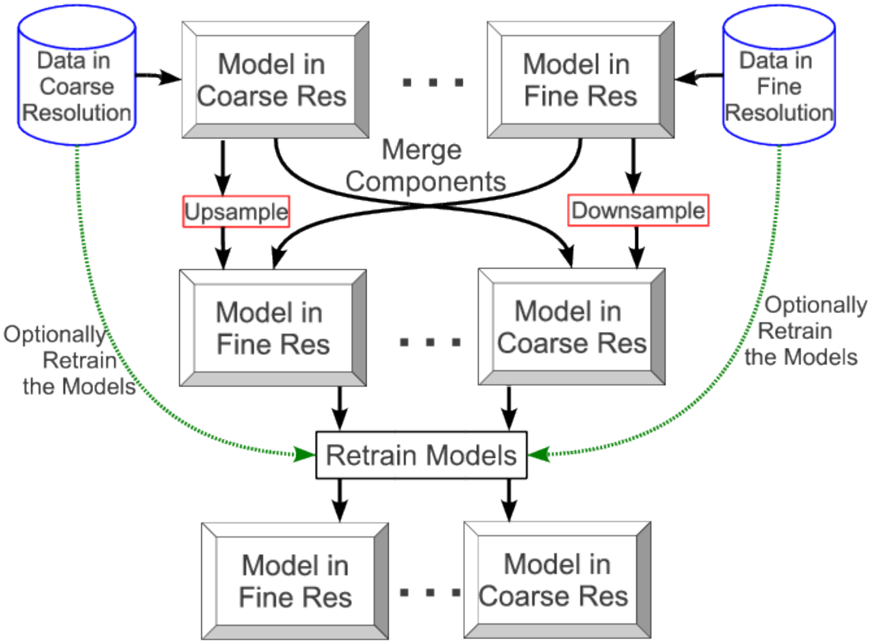

Figure 2: Multiresolution mixture modelling by merging of similar mixture components in different resolutions.

We experiment our proposed algorithm on the two chromosomal aberration patterns dataset in cancer genomics. One of the two datasets available from (Myllykangas et al., 2008; Hollmén and Tikka, 2007) was 393 dimensional (Coarse Resolution). In contrast, the other dataset available from (Baudis, 2007) was 862 dimensional (Fine Resolution). However, both the datasets explain the same phenomenon of chromosomal aberration and measure the similar chromosomal regions albeit in the different resolutions as explained in (Shaffer and Tommerup, 2005). The available datasets were converted to 0-1 matrix where each row denotes a cancer patient and each column denotes a region in the genome (a chromosome band). Since the chromosomal aberrations data had small number of samples (≈4500) and high dimensionality, we experimented chromosome-wise on the data as in (Tikka et al., 2007). When the data for each chromosome is extracted from the genome, the different chromosomes will have the different dimensionality and some rows contain only zeros. Such rows with only zeros were removed because they contain no information with respect to the aberration pattern in the cancer patient in that chromosome.

Figure 2 summarizes the experimental procedure showing that the mixture models in two different resolutions are learned separately using the EM algorithm (Dempster et al., 1977). We then calculate the symmetric KL divergence between the different components in the two mixture models and match the components that have the minimum KL divergence using the minimum weight bipartite matching (West, 1996). The similar components are merged using the Equations (5) and (6). Similarly, the parameters of the component distributions are merged using the Equation (7). The merging of the mixture components are performed repeatedly until the changes in the KL divergence between the two mixture models in any two iterations is small (e.g. less than 10^{-3} in our experiments).

The estimated the time to compute our approximation of the KL divergence and also that of the full KL divergence to show the performance improvement gained by our approximation of the KL divergence in (Adhikari and Hollmén, 2012) shows that our approximation is considerably faster than the full KL divergence.

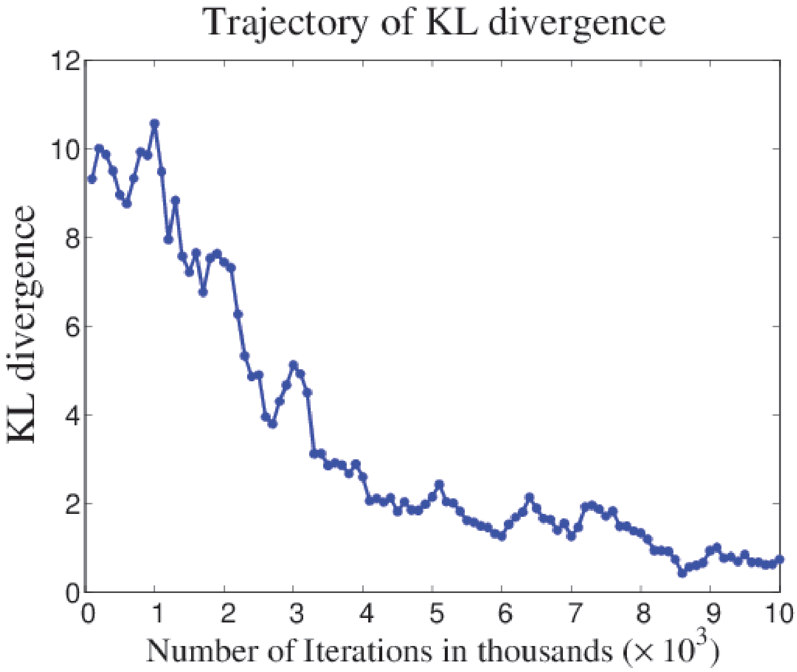

Figure 3: Changes in the KL divergence with increasing iterations.

Figure 3 shows the decrease in the KL divergence with the increasing iterations of minimizing the KL divergence. The Figure 3 is similar to convergence analysis as the KL divergence decreases with increasing number of iterations. Note in this case, the mixture model is not retrained. The decrease in the KL divergence is as expected not smooth as some merging of components in the two resolutions will make differences in both the models at different resolutions in the different iterations. The KL divergence between the two models will approach to zero when both the models are similar to each other after repeated merging. However, it is not exactly zero in our case because of the upsampling and downsampling of the parameters makes the two models not exactly equal.

We used the upsampling and downsampling of the model parameters as discussed in Section (4) to merge mixture components in two different resolutions (having different dimensionality). The merged mixture model can be optionally trained on the combined data via the EM algorithm (Dempster et al., 1977) initialized using the merged mixture model. However, retraining the mixture model in each iteration of minimizing the KL divergence is computationally inefficient and the initialization model does not considerably vary in each iteration thus producing final model that is not different from the original model. Therefore, we can optionally retrain the mixture model in every thousandth iteration.

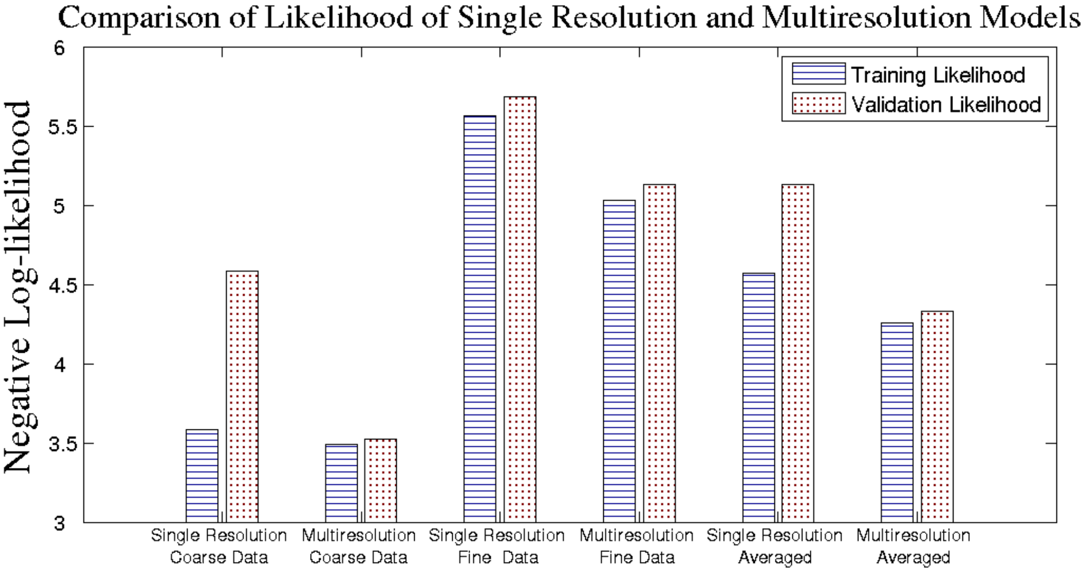

Figure 4: Likelihood of the multiresolution mixture model trained in a 10-fold cross-validation setting compared to the mixture model model in a single resolution. The result is an example case in the chromosome 17. Since the Y-axis shows the negative log-likelihood values, the shorter the bar the better the result.

We trained the multiresolution model and also the model in a single resolution in a ten-fold cross-validation setting via the EM algorithm. Here, the EM algorithm was initialized using the merged model in case of the multiresolution model and the single resolution models are initialized at random. Both the models were then trained to convergence via the EM algorithm. The multiresolution model is trained on the combined data obtained after integrating the data in two different resolution by transforming the data to the same resolutions as in Adhikari and Hollmén (2010b). However, the single resolution model were trained with data in only one resolution ignoring the data in other resolution which is the current state-of-the-art.

The results in the Figure 4 show that the likelihood of the multiresolution models are better than that of the models trained on the data in the single resolution. Since the Y-axis in the figure shows the negative log-likelihood, the shorter the bar better the result. The Figure 4 shows three different cases of the likelihood: single resolution model on the coarse and the fine data, multiresolution model on the coarse and the fine data; and finally average of likelihood in the coarse and the fine data by multiresolution and single resolution model. The performance of the multiresolution model is markedly better in the coarse resolution and only slightly better in the fine resolution because the number of samples in the dataset is very small in the coarse resolution to add more information to the model in the fine resolution. Nevertheless, the average likelihood by the multiresolution shows noticeably improved performance of the multiresolution mixture models in both the resolutions.

Since the ten-fold cross-validation produces only ten values of the likelihoods which are very small to perform statistical significance testing on the result, we performed hundred-fold cross-validation producing 100 different training and validation likelihood values each. With the 100 different likelihood values, we performed the the two-tailed t-test to ascertain the statistical significance of our result. The results show that both the validation and the training likelihoods are statistically significant when the significance level, α, is 0.1. Training likelihoods in both the coarse and the fine resolution as well as the training likelihood in coarse resolution is statistically significant when the significance level, α, is 0.05. The validation likelihood in fine resolution is not significant when the significance level, α, is 0.05. The p-value of the validation likelihood in fine resolution is 0.09. This result in fine resolution can be attributed to the fact that the number of samples that have been added to the combined data by upsampling from the coarse data is small (342 samples have been added compared to 2716 samples in the fine resolution in the chromosome 17). which is considerably less to substantially improve the performance of the multiresolution model.

In order to show that our strategy of merging the mixture components positively facilitates the training of the mixture model via the EM algorithm, we calculated the iterations required by the EM algorithm to converge when initialized using the merged model. The left panel in the Figure 5 shows the number of iterations required as an average over five different runs of merging and retraining the models to converge to the final model via the EM algorithm. From the figure, we can see that the number of iterations required for the EM algorithm to converge decreases as we increase the iterations for minimizing the KL divergence. The decrease in the number of iterations shows that merging the mixture components moves the mixture model closer to the final model with regards to training via the EM algorithm. For each number of iterations, we restart the minimization of KL divergence such that the EM algorithm is used to train the final mixture model only once.

6.1. Illustration using models in same resolution

We experimented our algorithm on the two models in the same resolution to ascertain the improvements in the performance of the proposed multiresolution mixture modelling algorithm. As shown in in the right panel of the Figure 5, we selected two models such that the one fits the data poorly (solid line with the circles) while the another model fits

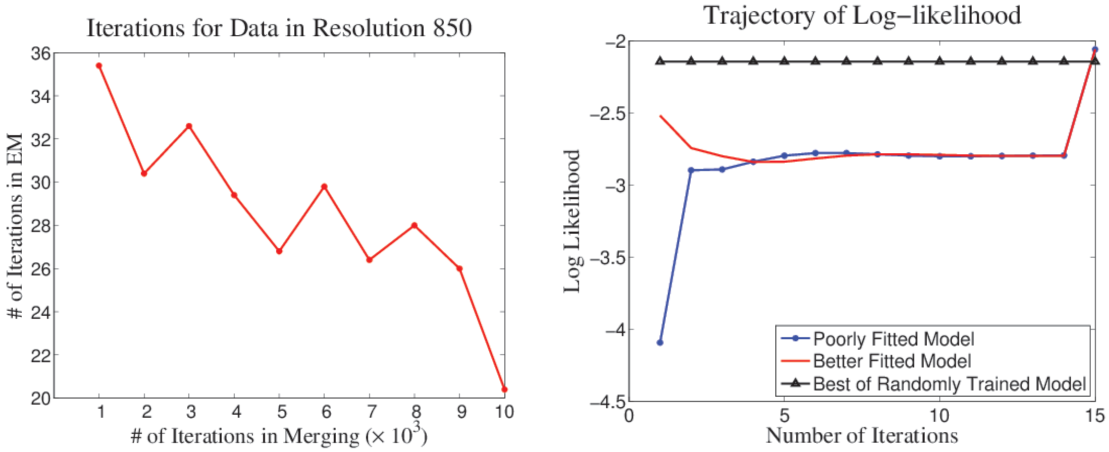

Figure 5: The left panel shows the number of iterations in the EM algorithm required to converge when the EM algorithm is initialized using the merged model after the minimization of KL is converged. The right panel shows the trajectory of the log likelihood values for two models in same resolution: a better fitted (solid line without the circles and the triangles) and a poorly fitted model (a solid line with the circles). The solid line with the triangles denotes the likelihood of the best of the 100 models trained on the combined data. The fifteenth iteration is the model retrained on the original data using the model after the 14th iteration as the initialized model. This is an example case in chromosome 21.

the data better (solid line without the circles and the triangles). We run our algorithm on the two models without upsampling and downsampling of model parameters as the number of model parameters are the same. The results obtained are visualized in the right panel of the Figure 5 showing that when the model converges, the likelihood on the combined data is better than the average likelihood of two models. However, the average likelihood is lesser than the best of randomly trained 100 models to convergence (solid line with triangles). Nevertheless, the likelihood obtained by merging the mixture components can be considered the validation likelihood as models are trained on the downsampled data and not on the combined data.

If we train the merged mixture model to convergence where the model obtained after the fourteenth iteration of merging is used to initialize the EM algorithm, we get the mixture model which produces better likelihood than the best of the randomly trained mixture model as shown in the fifteenth iteration in the right panel of the Figure 5. The improvement shows that our model is better than the best of the randomly trained model and also explains the importance of retraining the mixture model after every thousandth iteration. Furthermore, the fact that the merged mixture model producing better results experimentally verifies that our algorithm may be useful in avoiding local optima by making the little changes to the mixture models in the coarse and the fine resolution and also modelling the interactions across different resolutions. However, this result is neither mathematically proved and nor experimentally verified to work in every repeat of the experiment.

7. Summary and Conclusions

Multiresolution data arise when an object or a phenomenon is described at several levels of detail. In order to cope with the multiresolution data, standard data analysis methods need to be extended to include capabilities to model data in several resolutions simultaneously. In this paper, we proposed an algorithm for the multiresolution mixture modelling by merging the mixture components across the different resolutions. Given datasets in different resolutions from the same domain having similar distribution, our algorithm takes the models in different resolutions and repeatedly merges the mixture components minimizing the KL divergence thus producing a better mixture model. We performed experiments with our proposed algorithm on the two real-world chromosomal aberration datasets. The experiments show that our algorithm improves on the results of the competing multiresolution methods by involving the model interaction between the models in different resolutions.

Acknowledgments

This work has been funded by Helsinki Doctoral Programme in Computer Science - Advanced Computing and Intelligent Systems (Hecse), Helsinki Institute for Information Technology (HIIT), and Finnish Center of Excellence for Algorithmic Data Analysis Research (ALGODAN).

References

P. R. Adhikari and J. Hollmén. Preservation of statistically significant patterns in multiresolution 0-1 data. In Tjeerd Dijkstra, Evgeni Tsivtsivadze, Elena Marchiori, and Tom Heskes, editors, *Pattern Recognition in Bioinformatics*, volume 6282 of *Lecture Notes in Computer Science*, pages 86–97. Springer Berlin / Heidelberg, 2010a.

P. R. Adhikari and J. Hollmén. Patterns from multiresolution 0-1 data. In *Proceedings of the ACM SIGKDD Workshop on Useful Patterns*, UP '10, pages 8–16, New York, NY, USA, 2010b. ACM. ISBN 978-1-4503-0216-6.

P. R. Adhikari and J. Hollmén. Fast Progressive Training of Mixture Models for Model Selection. In J.-G. Ganascia, P. Lenca, and J.-M. Petit, editors, *Proceedings of Fifteenth International Conference on Discovery Science (DS 2012)*, volume 7569 of *Lecture Notes in Artificial Intelligence*, pages 145–156. Springer-Verlag, November 2012. ISBN 978-3-642-33491-7.

M. Baudis. Genomic imbalances in 5918 malignant epithelial tumors: An explorative meta-analysis of chromosomal CGH data. *BMC Cancer*, 7, 2007.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

V. Cristini and J. Lowengrub. *Multiscale Modeling of Cancer: An Integrated Experimental and Mathematical Modeling Approach.* Cambridge University Press, 2010.

I. Dagan, L. Lee, and F. Pereira. Similarity-based methods for word sense disambiguation. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, EACL '97, pages 56–63, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

M. A. R. Ferreira and H. K. H. Lee. *Multiscale modeling: a Bayesian perspective.* Springer series in statistics. Springer-Verlag, 2007. ISBN 9780387708973.

M. A. T. Figueiredo and A. K. Jain. Unsupervised Learning of Finite Mixture Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

R. Ghaemi, N. Sulaiman, H. Ibrahim, and N. Mustapha. A Survey: Clustering Ensembles Techniques. *Proceedings of World Academy Of Science, Engineering and Technology*, 38: 644–653, 2009. ISSN 2070-3740.

A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data*, 1(1), March 2007. ISSN 1556-4681.

J. Hollmén and J. Tikka. Compact and understandable descriptions of mixture of Bernoulli distributions. In M.R. Berthold, J. Shawe-Taylor, and N. Lavrač, editors, *Proceedings of the 7th International Symposium on Intelligent Data Analysis (IDA 2007)*, volume 4723 of *Lecture Notes in Computer Science*, pages 1–12, Ljubljana, Slovenia, September 2007. Springer-Verlag.

S. Kullback. *Information Theory and Statistics.* John Wiley and Sons, New York, 1959.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Y. Li and L. Li. A Novel Split and Merge EM Algorithm for gaussian mixture model. In *ICNC '09. Fifth International Conference on Natural Computation, 2009.*, volume 6, pages 479–483, August 2009.

S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

G. J. McLachlan and D. Peel. *Finite mixture models*, volume 299 of *Probability and Statistics – Applied Probability and Statistics Section.* Wiley, New York, 2000.

S. Myllykangas, J. Tikka, T. Böhling, S. Knuutila, and J. Hollmén. Classification of human cancers based on DNA copy number amplification modeling. *BMC Medical Genomics*, 1 (15), May 2008.

L. G. Shaffer and N. Tommerup. *ISCN 2005: An International System for Human Cytogenetic Nomenclature(2005) Recommendations of the International Standing Committee on Human Cytogenetic Nomenclature.* Karger, 2005.

P. Smyth. Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing*, 10:63–72, 2000. ISSN 0960-3174.

J. Tikka, J. Hollmén, and S. Myllykangas. Mixture Modeling of DNA copy number amplification patterns in cancer. In Francisco Sandoval, Alberto Prieto, Joan Cabestany, and Manuel Graña, editors, *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN 2007)*, volume 4507 of *Lecture Notes in Computer Science*, pages 972–979, San Sebastián, Spain, 2007. Springer-Verlag.

A. P. Topchy, A. K. Jain, and W. F. Punch. A Mixture Model for Clustering Ensembles. In M. W. Berry, U. Dayal, C. Kamath, and D. B. Skillicorn, editors, *Proceedings of the Fourth SIAM International Conference on Data Mining (SDM)*. SIAM, 2004.

N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM Algorithm for Mixture Models. *Neural Computation*, 12(9):2109–2128, 2000.

S. Vega-Pons and J. Ruiz-Shulcloper. A Survey of Clustering Ensemble Algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(3):337–372, 2011.

D. B. West. *Introduction to graph theory.* Prentice Hall, Second (Illustrated) edition, 1996.

A. S. Willsky. Multiresolution Markov models for signal and image processing. *Proceedings of the IEEE*, 90(8):1396–1458, August 2002. ISSN 0018-9219.

R. Wilson. MGMM: multiresolution Gaussian mixture models for computer vision. In *Proceedings of 15th International Conference on Pattern Recognition, 2000.*, volume 1, pages 212–215, 2000.

J. H. Wolfe. Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research*, 5:329–350, 1970.

Z. Zhang, C. Chen, J. Sun, and K. L. Chan. EM algorithms for Gaussian mixtures with split-and-merge operation. *Pattern Recognition*, 36(9):1973–1983, September 2003.