The Coreset Variational Bayes (CVB) Algorithm for Mixture Analysis

Qianying Liu[a], Clare A McGrory[a], Peter WJ Baxter[a,b]

[a] *Centre for Applications in Natural Resource Mathematics*
School of Mathematics, The University of Queensland
Brisbane, Australia, Qld 4072.
Tel.: +61 7 336 51385
Fax: +61 7 336 51477
[b] *Queensland University of Technology*
Brisbane, Australia, Qld 4001.

Abstract

The pressing need for improved methods for analysing and coping with big data has opened up a new area of research for statisticians. Image analysis is an area where there is typically a very large number of data points to be processed per image, and often multiple images are captured over time. These issues make it challenging to design methodology that is reliable and yet still efficient enough to be of practical use. One promising emerging approach for this problem is to reduce the amount of data that actually has to be processed by extracting what we call coresets from the full dataset; analysis is then based on the coreset rather than the whole dataset. Coresets are representative subsamples of data that are carefully selected via an adaptive sampling approach. We propose a new approach called coreset variational Bayes (CVB) for mixture modelling; this is an algorithm which can perform a variational Bayes analysis of a dataset based on just an extracted coreset of the data. We apply our algorithm to weed image analysis.

Keywords: Mixture modelling, Coresets, Variational Bayes, Image Analysis, Bayesian Statistics

1. Introduction

Finite mixture models are applied in diverse areas of science and provide a straightforward, but flexible extension of classical parametric models (Fruhwirth-Schnatter, 2006). They are used in situations where it is

Email addresses: qianying.liu@uq.net.au (Qianying Liu), c.mcgrory@uq.edu.au (Clare A McGrory), p.baxter@qut.edu.au (Peter WJ Baxter)

thought that there is more than one sub-population giving rise to points in the dataset. The sub-populations are each represented by one of the components that comprise the mixture. An obvious scenario where this type of model would be suitable is image analysis: we can associate ranges of intensity level present in the image with components of the mixture.

We take a Bayesian approach to our inference. In Bayesian analysis the posterior distribution of the unknown parameters can be very difficult to estimate. The most commonly used Bayesian method for estimating the parameters is Markov chain Monte Carlo (MCMC), e.g. a Gibbs sampling approach (Alston et al., 2012). However, while MCMC-based approaches are very accurate, the computational requirements associated with this approach can be prohibitive for very large datasets. Less computationally demanding algorithms for fitting the mixture models are alternative approximate Bayesian inference techniques such as variational Bayes (VB) (McGrory & Titterington, 2007; Ormerod & Wand, 2010) and approximate Bayesian computation (ABC) (Marin et al., 2012).

Even though the standard VB method is highly computationally efficient in comparison to MCMC, in some cases it still might be too time-consuming for very large data problems if analysis is required within short time-frames. Consider for instance the weed-crop imaging application that we will explore in this article. Images have an extremely large number of pixels, there will likely be multiple images to analyse, and estimates are needed reasonably quickly. In order to achieve this, time-efficient techniques are required. Another avenue is to reduce the volume of data that actually has to be processed in the fist place. Removing part of the dataset before running the analysis is a less drastic thing to do than we might think when we consider that much of the dataset gives us the same or very similar information. For example, consider trying to fit a mixture model to an image, we do not necessarily have to analyse all of the observed intensity levels present in the full dataset to obtain a good estimate of the mean for that particular cluster. The coreset approach (Feldman et al., 2011) is a data-reduction algorithm. It involves extracting a representative subsample of the dataset which can then be analysed in order to make inference about the whole dataset. In Feldman et al. (2011) the coreset approach was used within a classical mixture modelling framework with good results. In Ahfock et al. (2014) the Gibbs sampler was modified for use with coresets; the resulting algorithm was called the weighted Gibbs sampler. This was shown to give very good results when applied to analysing satellite image data and it drastically reduced the computation time required for the analysis. In a similar spirit, we wish to modify the VB algorithm to make it suitable for use with coresets of data. Since VB is more time-efficient than the Gibbs sampler, combining the concept of coresets with VB will result in even greater reductions in computational burden. We refer to this new algorithm as coreset variational Bayes (CVB).

It could be argued that a spatial mixture model is more appropriate than a finite mixture model for modelling a dataset such as an image which contains spatial information (see e.g. McGrory et al. (2012)) as this might slightly increase the clustering accuracy. The reason we do not incorporate a spatial component into our modelling is that we cannot afford the huge extra computational burden this would incur since we require a very time efficient approach for big data settings. In this way there is a trade-off between these two aspects.

Weeds are defined as being plants which have originated in and continue to evolve in a natural environment; but they are problematic because they do so in a manner which interferes with the growth of crops or other agriculture related activities (Zimdahl, 2009). Weeds are generally better able to compete than crops for resources like minerals, light and water. This leads to a lot of waste of agricultural investment in cases where weed plants are present, because most of these valuable inputs would be used up by them instead of by the valuable crops. There are many other harmful problems associated with weeds, such as the harbouring of extra pests in the area which can cause plant diseases that may infect the crops in the region. Hence, it is important to be able to effectively manage weeds in farming regions if a nation's agriculture industry is to remain competitive in international markets (Sindenab et al., 2004).

A growing area of research that has the potential to be very useful in weed management is the use of statistical methodology to analyse images of weeds used in agriculture trials. This might be of use in projects where the aim is to assess and compare the effectiveness of different chemical weed killers for instance. If calculation of the proportion of a plot of land that is weed, soil or plant after chemical applications can be done by analysing an image, this will save time by removing the need for researchers to go out into the fields and count the live and dead plants by hand. Due to the large number of pixels present in a typical image, and the fact that there will most likely be multiple images to analyse in a given trial, it is important to find a very time efficient algorithm for processing this type of data (see e.g. Kargar and Shirzadifar (2013)).

2. Finite mixture model

Mixture models provide an excellent and flexible way to represent complex distributions. The mixture model we fit to our data comprises a linear combination of standard mixture models, these are called the components of the mode. Each component has a corresponding mixture weight which reflects the expected proportion of the data that might be captured by that particular component. In our finite mixture model for a set of continuous observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, assume that the observations are all generated *i.i.d.* (independent and identically distributed) from a random variable $\boldsymbol{Y}$, which follows a mixture of K independent Gaussian distributions. In the missing-data interpretation of the mixture we introduce an unobserved indicator variable $\boldsymbol{z}_{ij}$ for each observation; this identifies the component allocations of our observations by taking value 1 if observation i is from component j, and 0 otherwise. Since these indicators are unknown to us, the model is therefore a missing-data model. For each component, j, the value of ρ_j is the relevant weight for that particular jth component. The model density is given by:

$$p(\mathbf{y}, \boldsymbol{z}|\boldsymbol{\theta}) = \prod_{i=1}^{n}\prod_{j=1}^{K}\left\{\rho_j N_d\left(\mathbf{y}_i; \boldsymbol{\mu}_j, \boldsymbol{T}_j^{-1}\right)\right\}^{z_{ij}}.$$

Here, $N_d(\cdot,\cdot)$ represents the d-dimensional multivariate normal density, where $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K)$ and $\boldsymbol{T} = (\boldsymbol{T}_1, \cdots, \boldsymbol{T}_K)$, and $\boldsymbol{T}_j$ denotes the jth precision matrix, which is the inverse of the jth covariance matrix.

3. Bayesian priors

We follow the standard Bayesian conjugate prior setting (Alston et al., 2012) for this model, and choose hyper-parameters such that they correspond to non-informative prior settings, thus allowing information contained in the dataset to have more influence over the fit. The weight coefficients are assigned Dirichlet prior distributions:

$$p(\boldsymbol{\rho}) = Dir(\boldsymbol{\rho}; \alpha_1^{(0)}, \cdots, \alpha_K^{(0)}).$$

The prior distributions of the means conditioned on the covariance matrices are independent multivariate normal distributions:

$$p(\boldsymbol{\mu}|\boldsymbol{T}) = \prod_{j=1}^{K} N_d\left(\boldsymbol{\mu}_j\,;\boldsymbol{m}_j^{(0)}, (\beta_j^{(0)}\boldsymbol{T}_j)^{-1}\right).$$

The prior of the precision matrices are given by Wishart distributions:

$$p(\boldsymbol{T}) = \prod_{j=1}^{K} W\left(\boldsymbol{T}_j\,;\boldsymbol{v}_j^{(0)}, \boldsymbol{\Sigma}_j^{(0)}\right).$$

Therefore, the joint distribution would finally be:

$$p(\boldsymbol{y},\boldsymbol{z},\boldsymbol{\theta}) = p(\boldsymbol{y},\boldsymbol{z}\,|\,\boldsymbol{\theta})p(\boldsymbol{\rho})p(\boldsymbol{\mu}|\boldsymbol{T})p(\boldsymbol{T}).$$

The quantities of $\{\alpha_j^{(0)}\}, \{\boldsymbol{m}_j^{(0)}\}, \{\beta_j^{(0)}\}$ and $\{\boldsymbol{\Sigma}_j^{(0)}\}$ are all hyper-parameters.

4. Bayesian posterior distributions

4.1. The variational approach

The variational Bayesian method is a time-efficient approach for estimating Bayesian mixture models (Faes et al., 2011; McGrory & Titterington, 2007; Wand et al., 2012) and can be thought of as an alternative to MCMC. The main difference between the two approaches is that instead of estimating the parameter directly by sampling from the posterior distribution, variational Bayesian methods approximate it. This is done by artificially ”introducing” a more amenable distribution $q(\boldsymbol{\theta},\boldsymbol{z})$ which is often referred to as the variational approximating function. This function will end up becoming an approximate the joint conditional distribution of $\boldsymbol{\theta}$ and $\boldsymbol{z}$ given the observations $\mathbf{y}$ after the variational method is applied to it. In the following we explain how $q(\boldsymbol{\theta},\boldsymbol{z})$ should be chosen and integrated into the variational framework in order to achieve this outcome.

The distribution $q(\boldsymbol{\theta},\boldsymbol{z})$ is chosen to minimise the Kullback-Leibler(KL) divergence between the approximating density $q(\boldsymbol{\theta},\boldsymbol{z})$ and the true joint density $p(\boldsymbol{\theta},\boldsymbol{z}\,|\,\boldsymbol{y})$. In doing so, we are trying to obtain a relatively tight lower bound on the marginal density, $p(\boldsymbol{y})$. Essentially we manipulate and re-express the joint density to allow us to introduce the variational approximating function in such a way that we can then use a maximisation approach to estimate parameters of that target approximating function (see also McGrory

& Titterington (2007)). We begin then by showing that the joint density is lower bounded as:

$$\begin{aligned}\log p(\boldsymbol{y}) &= \log \int \sum_{\{\boldsymbol{z}\}} p(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \log \int \sum_{\{\boldsymbol{z}\}} q(\boldsymbol{\theta}, \boldsymbol{z}) \frac{p(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\theta})}{q(\boldsymbol{\theta}, \boldsymbol{z})} d\boldsymbol{\theta} \\ &\geq \int \sum_{\{\boldsymbol{z}\}} \log \frac{p(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\theta})}{q(\boldsymbol{\theta}, \boldsymbol{z})} d\boldsymbol{\theta}. \quad \text{by Jensen's inequality} \qquad (1)\end{aligned}$$

Another way of viewing this is that finding the tightest lower bound is the same as minimising the Kullback-Leibler divergence between the variational distribution and the true target posterior. It is exactly minimised when we take $q(\boldsymbol{\theta}, \boldsymbol{z}) = p(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{y})$. However, as we are trying to simplify the problem, $q(\boldsymbol{\theta}, \boldsymbol{z})$ should be a close enough approximation to the true density, yet have a simple form for computational purposes. Normally, to achieve this $q(\boldsymbol{\theta}, \boldsymbol{z})$ is restricted to have the factorised form $q(\boldsymbol{\theta}, \boldsymbol{z}) = q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) q_{\boldsymbol{z}}(\boldsymbol{z})$.

Unlike the MCMC approach, the variational method approximates the parameters in the finite mixture model. This difference may cause a slight decrease in accuracy of the variational method when compared with MCMC. It has been demonstrated that in many contexts, including mixture modelling (e.g. McGrory & Titterington (2007); Wand et al. (2012); Faes et al. (2011)) that the variational method can largely reduce operating time and yet the loss in accuracy that arises from the approximation is not terribly great. Indeed the approximate result is typically adequate for practical purposes. When computational efficiency is an important consideration, it is worthwhile for us to pursue this approximate approach rather than MCMC.

4.2. Variational posterior

After we maximise the lower bound (equation (1)), the posteriors are

$$q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) = Dir\left(\boldsymbol{\rho}; \alpha_1, \cdots, \alpha_k\right),$$

$$q_{\boldsymbol{\mu}|\boldsymbol{T}}(\boldsymbol{\mu}|\boldsymbol{T}) = \prod_{j=1}^{K} N_d\left(\boldsymbol{\mu}_j; \boldsymbol{m}_j, (\beta_j \boldsymbol{T}_j)^{-1}\right),$$

and

$$q_{\boldsymbol{T}}(\boldsymbol{T}) = \prod_{j=1}^{K} W\left(\boldsymbol{T}_j; \boldsymbol{v}_j, \boldsymbol{\Sigma}_j\right),$$

the hyperparameters will be updated as:

$$\begin{aligned}
\alpha_j &= \alpha_j^{(0)} + \sum_{i=1}^{n} q_{ij} && (2)\\
\beta_j &= \beta_j^{(0)} + \sum_{i=1}^{n} q_{ij} && (3)\\
\boldsymbol{m}_j &= \frac{\beta_j^{(0)}\boldsymbol{m}_j^{(0)} + \sum_{i=1}^{n} q_{ij}\boldsymbol{y}_i}{\beta_j} && (4)\\
\boldsymbol{\Sigma}_j &= \boldsymbol{\Sigma}_j^{(0)} + \sum_{i=1}^{n} q_{ij}\boldsymbol{y}_i\boldsymbol{y}_i^T + \beta_j^{(0)}\boldsymbol{m}_j^{(0)}\boldsymbol{m}_j^{(0)^T} - \beta_j\boldsymbol{m}_j\boldsymbol{m}_j^T && (5)\\
v_j &= v_j^{(0)} + \sum_{i=1}^{n} q_{ij}, && (6)
\end{aligned}$$

with q_{ij} being the variational posterior expected probability that the indicator variable $z_{ij} = 1$. The form of q_{ij} is:

$$\begin{aligned}
q_{ij} &= \frac{\exp\left\{\langle \log \rho_j \rangle + \frac{1}{2}\{\langle \log|\boldsymbol{T}_j|\rangle\}\right\} - \frac{1}{2}tr\left(\langle \boldsymbol{T}_j\rangle(\boldsymbol{y}_i - \boldsymbol{m}_j)(\boldsymbol{y}_i - \boldsymbol{m}_j)^T + 1/\beta_j\right)\mathbf{I}_d}{s_i}\\
&= \frac{\phi_{ij}}{s_{ij}}, \qquad (7)
\end{aligned}$$

where s_{ij} is the normalization constant and $\langle\cdot\rangle$ denotes the expected values required to evaluate the expressions in Equation 7. These are given by:

$$\begin{aligned}
&\langle \boldsymbol{\mu}_j \rangle = \boldsymbol{m}_j,\\
&\langle \boldsymbol{T}_j \rangle = v_j\boldsymbol{\Sigma}_j^{-1}\\
&\langle |\log \boldsymbol{T}_j|\rangle = \sum_{s=1}^{d}\Psi\left(\frac{v_j + 1 - s}{2}\right) + d\log(2) + \log|\boldsymbol{\Sigma}_j|\\
&\langle |\log(\rho_j)\rangle = \Psi(\hat{\alpha}_j) + \Psi(\hat{\alpha}_{\boldsymbol{.}}),
\end{aligned}$$

where $\Psi(\boldsymbol{.})$ is the digamma function and $\hat{\alpha}_{\boldsymbol{.}} = \sum_j \hat{\alpha}_j$.

4.3. The Standard Variational Bayes (VB) algorithm

In Algorithm 1 we outline the pseudo-code for the VB algorithm for obtaining the posterior estimates. We draw the reader's attention to what we call the component elimination property of VB. By this we mean that the algorithm can automatically determine the complexity of the model since:

1. the initial value for the number of components is set to be greater than what the user would reasonably expect in the final fit,
2. components which converge to have similar estimated parameter values will be dominated by only one of them, then components with small mixing weights can be removed leading to automatic complexity assessment.

This feature is very useful in applications involving large amounts of data as it greatly reduces computing costs: no need to perform different runs for various numbers of mixture components and compare them, which is what we would have to do if we used an MCMC approach in order to estimate the dimension. The other alternative would be to use the computationally burdensome reversible jump MCMC. This feature is also particularly useful for practical applications where the operator would like the analysis to run in an unsupervised manor because there is no need for the user to manually control searches over dimensionality.

Of course initial settings for the hyperparameters and epsilon (the threshold for component removal) can have some influence on estimation some cases, although previous research has shown the method to be generally fairly robust to initial parameter settings. We chose values for the hyperparameters $\alpha^{(0)}, \beta^{(0)}, \mathbf{\Sigma}^{(0)}, v^{(0)}, \boldsymbol{m}^{(0)}$ to correspond to vague non-informative priors. Note that these are the hyperparameters of the Gaussian mixture model, and they need to be chosen in the initial settings of the variational algorithm. In this we follow the standard guidelines for prior settings used in any Bayesian analysis. Naturally, if a user has specific prior knowledge in their particular application, they might decide an informative prior is suitable in their case, this is a user driven choice.

Note that the choice of epsilon determines how small a component's allocated weighting has to be at a given iteration of the algorithm in order for it to be selected for removal, and again is the user's choice, within reason of course. The simplest, and perhaps "safest" choice is to set that equal to 1. That is a "safe" option because clearly once number of allocations is less than 1, the user can be sure that they are not excluding a component that has any real significant contribution within the model. However, that might not be the most time efficient choice, if you imagine a dataset with thousands of observations, it might be that any component with less than say 50-100 observations allocated is unlikely to be useful and on the way to being eventually removed with a gradual reduction in allocations at each subsequent iteration. However, choosing epsilon in the range 50-100 might be less "safe" because it does incur higher risk that a component that is in fact potentially useful in representing some features of the data will get removed in error. Some users might decide on a proportional value of the size of the observation set which can be used as the cut off, e.g. 1% of the dataset size. Users must select a suitable value according to their particular application.

Algorithm 1: The standard VB algorithm

```
Set initial number of components $K$.
Set initial values for hyperparameters $\alpha^{(0)}, \beta^{(0)}, \mathbf{\Sigma}^{(0)}, \upsilon^{(0)}, \boldsymbol{m}^{(0)}$.
Specify initial allocation of observations to components and get $q_{ij}$.
while Not Converged do:
    Update variational posterior expressions for model parameters: equations 2-6
    Update variational posterior for $q_{ij}$: equation 7
    if any component has a mixing weight$\leq \varepsilon$
        remove the component from model
    end if
    if the algorithm has converged
        exit loop.
    end if
end while
```

5. Adapting the Variational Bayes Approach for use with Coresets of Data

In Ahfock et al. (2014) it was shown how the Gibbs sampler could be adapted for use with coresets of data. In a similar spirit we will adapt the VB method for use with coresets. We first describe the basic procedure for finding coresets, as outlined in Feldman et al. (2011).

5.1. Finding coresets

The coreset method described in Feldman et al. (2011) can be used to find an appropriate weighted subset to represent the information in the complete dataset. The starting point is to first sample uniformly a small number of points, then remove half of the data points which are closest to the sampled points. Next sample again from the rest of the points and remove half of the points lying closest until all of the data points are labeled as removed or sampled.

By doing this, we construct a hierarchy of data points and the importance-weight of the sampled points is associated with the log-likelihood. The weights are set to be optimal if the estimated log-likelihood is of the least variance. This construction of a sampled set gives a higher probability to observations that are further away from the initial cluster center, and the sampling bias would be fixed by adapting the weight which is to be associated with the sampling probabilities. We can then finally build an appropriate coreset from the whole dataset based on the weights.

This algorithm for coreset construction is more formally summarised in pseudo-code form in Algorithm 2.

```
Algorithm 2: Algorithm for finding a coreset (see Feldman et al. (2011))
input: Whole dataset $\boldsymbol{D}, \epsilon, \delta, K$.
set: $\boldsymbol{D}' \leftarrow \boldsymbol{D}; \boldsymbol{B} \leftarrow \varnothing$
Specify initial allocation of observations to components.
while $|\boldsymbol{D}'| \geq 10dK\ln(1/\delta)\epsilon$ do :
    Sample set $\boldsymbol{S}$ of $\beta = 10dK\ln(1/\delta)$ points uniformly at random from $\boldsymbol{D}'$;
    Remove $\lceil|\boldsymbol{D}'|/2\rceil$ points $\boldsymbol{x} \in \boldsymbol{D}'$ closest to $\boldsymbol{S}$ (i.e., minimising dist $(\mathbf{x}, S)$) from $\boldsymbol{D}'$;
    Set $\boldsymbol{B} \leftarrow \boldsymbol{B} \cup \boldsymbol{S}$;
Set $\boldsymbol{B} \leftarrow \boldsymbol{B} \cup \boldsymbol{D}'$

for every $b \in \boldsymbol{B}$ do
    $\boldsymbol{D}_b \leftarrow$ the points in $\boldsymbol{D}$ whose closest point $\boldsymbol{B}$ is $b$.
for every $b \in \boldsymbol{B}$ and every $\mathbf{x} \in \boldsymbol{D}_b$ do
    $m(\mathbf{x}) \leftarrow \lceil \frac{5}{|D_b|} + \frac{\mathrm{dist}(\mathbf{x},\boldsymbol{B})^2}{\sum_{\mathbf{x}'\in\boldsymbol{D}} \mathrm{dist}(\mathbf{x},\boldsymbol{B})^2} \rceil$;
Pick a non-uniform random sample $\boldsymbol{C}$ of $10\lceil dk|\boldsymbol{B}^2\ln(1/\delta)/\epsilon^2\rceil$ points from $\boldsymbol{D}$,
where for every $\mathbf{x}' \in \boldsymbol{C}$ and $\mathbf{x}' \in \boldsymbol{D}$, we have $\mathbf{x}' = \mathbf{x}$ with probability
    $m(\mathbf{x})/\sum_{\boldsymbol{x}'\in\boldsymbol{D}} m(\boldsymbol{x})$;
for each $\mathbf{x}' \in \boldsymbol{C}$ do $\gamma(\mathbf{x}') \leftarrow \frac{\sum_{\boldsymbol{x}\in\boldsymbol{D}} m(\boldsymbol{x})}{|\boldsymbol{C}|\cdot m(\boldsymbol{x}')}$
output: Coreset $C = \left\{(\gamma(\mathbf{x_1}), \mathbf{x_1}), (\gamma(\boldsymbol{x_2}), \boldsymbol{x_2}), \ldots, (\gamma(\boldsymbol{x_{|C|}}), \boldsymbol{x_{|C|}})\right\}$.
```

5.2. VB inference using coreset sampling

In this section we propose a new algorithm in which we adapt the variational Bayes method in order that it can be used in conjunction with a coreset sampling approach. We will use the standard prior settings as described in section 3 and the posterior of the model will be adjusted using the coreset samples and coreset weights. This novel modification of the algorithm makes it suitable for use in analysing a coreset of the image. The update equations that have to be adapted for use with the coreset data are equations 2-7. In those equations $\boldsymbol{y}_i$ is replaced by $\hat{\boldsymbol{y}}_i$, where $\hat{\boldsymbol{y}}_i$ corresponds to the weighted observations and is defined as follows:

$$\hat{\boldsymbol{y}}_i = \frac{\gamma_i \times \boldsymbol{y}_i}{\frac{1}{n}\sum_{i=1}^{n} \gamma_i}.$$

The expected values required to update the expressions remain unaltered from the form they take in the standard VB algorithm. Pseudo-code for the weighted VB algorithm which we call coreset variational Bayes (CVB) is outlined in Algorithm 3.

Algorithm 3: Coreset Variational Bayes (CVB) Algorithm

```
input: $C = \{(\gamma(\mathbf{x_1}), \mathbf{x_1}), (\gamma(\mathbf{x_2}), \mathbf{x_2}), \ldots, (\gamma(\mathbf{x_N}), \mathbf{x_N})\}$ from Algorithm 2.
       Set initial values for hyper-parameters, $\varepsilon, \alpha^{(0)}, \beta^{(0)}, \boldsymbol{\Sigma}^{(0)}, v^{(0)}, m^{(0)}$.
       Specify initial allocation of observations to components via initial $q_{ij}$.
while Not Converged, do:
    Update modified variational posterior expressions for model parameters
    Update modified variational posterior for $q_{ij}$
    if any component has a mixing weight$\leq \varepsilon$
       remove the component from model
    end if
    if the algorithm has converged
       exit loop.
    end if
end while
```

6. Application to Weed-Crop Image Segmentation Using Coreset Variational Bayes (CVB)

6.1. Weed Image

We illustrate the effectiveness of the CVB algorithm for analysing an image of a weed plant amongst soil and dead leaves, see Fig 1. The size of the coreset is 2599 pixels, which is 1/100 of the size of the original dataset. The idea is to use the CVB algorithm to segment the image and classify the pixels as representing either living plant, background soil, or dead leaves. This is the type of classification that would be required for research purposes in agricultural trials. In running the analysis, the hyper-parameters are chosen to correspond to vague prior settings.

Figure 2 shows the segmentation of each component into the three different types of matter. As we can see, the area of the living part of the weed is clearly defined, and the algorithm can also distinguish between soil and dead leaves. The numerical segmentation results are in Table 1 and for comparison we also show the results that we would have obtained from an analysis of the full dataset. There is close agreement between these results showing that the coreset modification of the VB algorithm is reliable and useful. Significantly, the VB coreset algorithm is greatly more time-efficient than the standard VB as it runs around 18 times faster (around 40 minutes for CVB compared to around 12 hours for VB on the full dataset). This is very impressive when we consider how similar the final results were. Hence CVB is a practical and useful alternative to standard VB for the image segmentation based on finite Gaussian mixture models.

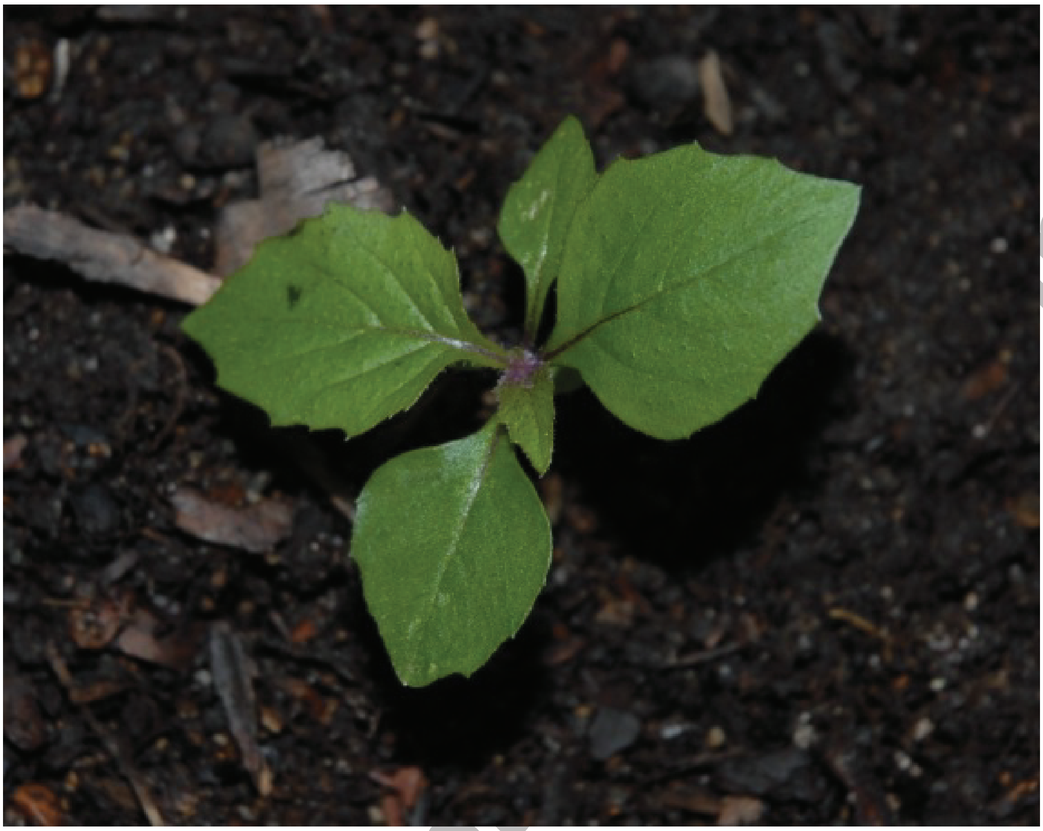

Figure 1: The original image of a weed against background soil to be analysed.

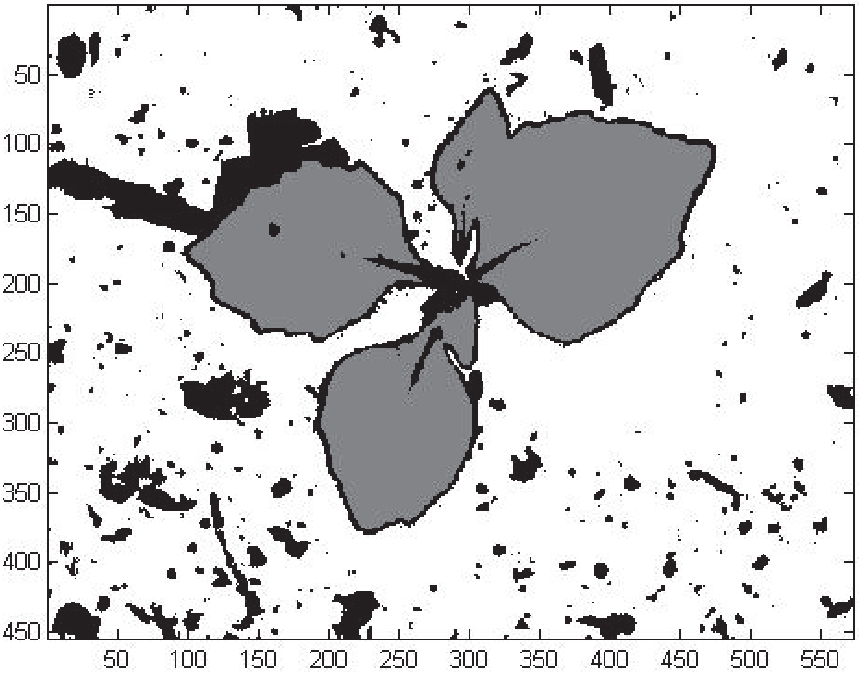

Figure 2: Result of the CVB analysis of the weed image. The pixels in the image have been segmented into 3 different components representing the types background soil, plant and dead leaf.

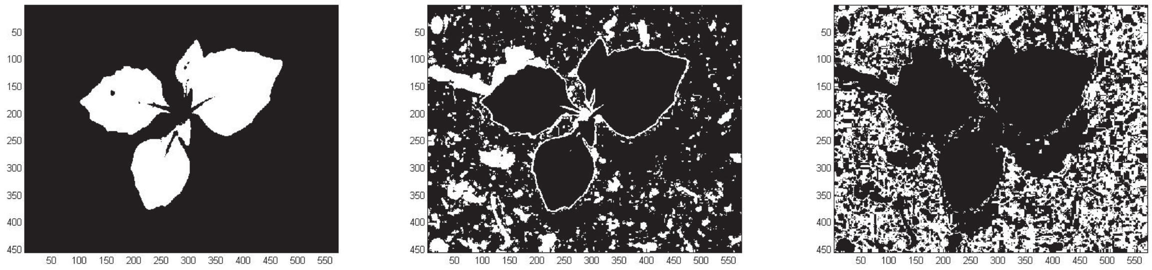

Figure 3: The three different components identified via the CVB algorithm shown in three separate plots. The first plot shows, in white, the area of living plant (weed), the second shows dead leaves. The third plot shows regions of background soil.

Table 1: Result comparison

Component	standard VB		VB coreset	
type	mixing weight	mean	mixing weight	mean
background soil	0.7005	$\begin{bmatrix} 0.1075 \\ 0.0986 \\ 0.0941 \end{bmatrix}$	0.6989	$\begin{bmatrix} 0.1093 \\ 0.0999 \\ 0.0952 \end{bmatrix}$
plant	0.1765	$\begin{bmatrix} 0.3200 \\ 0.4396 \\ 0.1753 \end{bmatrix}$	0.1894	$\begin{bmatrix} 0.3233 \\ 0.4422 \\ 0.1810 \end{bmatrix}$
dead leaves	0.1231	$\begin{bmatrix} 0.2660 \\ 0.2447 \\ 0.1986 \end{bmatrix}$	0.1117	$\begin{bmatrix} 0.2768 \\ 0.2561 \\ 0.2059 \end{bmatrix}$

7. Discussion

We have presented a new algorithm (CVB) for analysing data using variational Bayes based on a representative coreset of the data. This allows us to perform reliable inference in a highly time-efficient way. However, the running time after algorithm modification is still around 40 minutes for a medium size (67KB) image. While this is good in comparison to other existing approaches, there is still much scope for further research if these ideas are to be put to routine use. Consider that agricultural activities may need weed detection means applied on a large area of land, and thus the image can be more than 1GB, and contain even more detail than the presented example. One option to explore might be looking at improving computational speed through the use of more efficient programming algorithms, more sophisticated computers or GPU programming for increased efficiency Suchard et al. (2010). This is a topic for future research.

References

Ahfock, D., Alston, C.L., Horsley, J., and McGrory, C.A. (2014) Weighted Gibbs Sampler for Mixtures. *Stat*, 65.

Alston, C., Mengersen, K., and Pettitt, A.N. (2012) *Case Studies in Bayesian Statistical Modelling and Analysis*, 1st ed., John Wiley & Sons.

Faes, C., Ormerod, J., and Wand, M. (2011) Variational Bayesian Inference for Parametric and Nonparametric Regression with Missing Data. *J. Amer. Statist. Assoc.,* 106, 959–971.

Feldman, D., Faulkner, M., and Krause, A. (2011) Scalable Training of Mixture Models via Coresets, In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011*, (eds J. Shawe-Taylor, R.S, Zemel, P., Bartlett, F. Pereira, F and K.Q. Weinberger, KQ), 2142–2150. NY: Curran Associates, Inc.

Fruhwirth-Schnatter, S. (2006) *Finite mixture and Markov switching models.* New York: Springer.

Kargar, A.H.B. and Shirzadifar, A.M. (2013) Automatic weed detection system and smart herbicide sprayer robot for corn fields, In *Proceeding of the 2013 RSI/ISMInternational Conference on Robotics and Mechatronics February 13-15, Tehran, Iran.*

Marin, J.M., Pudlo, P., Robert,C.P., and Ryder, R. (2012) Approximate Bayesian Computation methods. *Statistics and Computing*, 22, 1167–1180.

McGrory, C.A., Pettitt, A.N., Reeves, R., Griffin, M., and Dwyer, M. (2012) Variational Bayes and the Reduced Dependence Approximation for the Autologistic Model on an Irregular Grid with Applications. *J. Comput. Graph. Stat.*, 21, 781–796.

McGrory, C.A., and Titterington, D.M. (2007) Variational approximations in Bayesian model selection for finite mixture distributions. *Comput. Stat. Data Anal.*, 51, 5352–5367.

Ormerod, J.T., and Wand, M.P. (2010) Explaining variational approximations. *Am. Stat.*, 64, 140–153.

Sindenab, J., Jonesbc, R., Hesterba, S., Odomba, D., Kalischda, C., Jamese, R., and Cacho, O. (2004) The economic impact of weeds in Australia, CRC for Australian Weed Management Technical Series no. 8.

Suchard, M.A., Wang, Q., Chan, C., Frelinger, J., Cron, A.J., and West, M. (2010) Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures. *J. Comput. Graph. Stat.*, 19, 419–438.

Wand, M., Ormerod, J., Padoan, S., and Fruhwirth, R. (2012) Mean Field Variational Bayes for Elaborate Distributions. *Bayesian Anal.,* 6, 847–900.

Zimdahl, R. (2009) *Fundamentals of Weed Science*. San Diego: Academic Press.