

README (Variational approximation for mixtures of linear mixed models)

Siew Li Tan and David J. Nott

The R code for the variational greedy algorithm (VGA) using Algorithms 1, 2 and 3 are included as part of the supplemental materials.

- To use VGA with Algorithm 1 where there is no hierarchical centering, run the code in ‘VGA Alg 1 (no centering).R’ and use the function

`greedy(y,X,W,V,U,vni,epm,M,IGa,IGb,sigdelp,sigbetap).`

- To use VGA with Algorithm 2 when only $X_i = W_i$ (partial centering), run the code in ‘VGA Alg 2 (partial centering).R’ and use the function

`greedyXW(y,X,W,V,U,vni,epm,M,IGa,IGb,sigdelp,sigbetap).`

- To use VGA with Algorithm 3 when $X_i = W_i = V_i$ (full centering), run the code in ‘VGA Alg 3 (full centering).R’ and use the function

`greedyXWV(y,X,W,V,U,vni,epm,M,IGa,IGb,sigdelp,sigbetap).`

The arguments are as described below:

y	is the $N \times 1$ vector, $y = (y_1, \dots, y_n)$
X	is the $N \times p$ matrix, $X = (X_1^T, \dots, X_n^T)^T$
W	is the $N \times s_1$ matrix, $W = (W_1^T, \dots, W_n^T)^T$
V	is the $N \times s_2$ matrix, $V = (V_1^T, \dots, V_n^T)^T$
U	is the $n \times d$ matrix, $U = (u_1, \dots, u_n)^T$
vni	is the $n \times 1$ vector, $vni = (n_1, \dots, n_n)$
M	is the number of attempts to split up each component in the current mixture model (we used $M = 5$ throughout the paper)
IGa IGb	refers to the hyperparameters α and β respectively of the inverse gamma prior $IG(\alpha, \beta)$. We used the same inverse gamma priors for $\sigma_{a_j}^2$, $\sigma_{b_j}^2$, $j = 1, \dots, k$ and σ_{jl}^2 , $j = 1, \dots, k$, $l = 1, \dots, g$, fixing IGa as 2 and suggest taking IGb as <code>IGb <- 2*(fitdistr(lm(y~X-1)\$residuals, 't', m=0, df=4)\$estimate)^2</code>
sigdelp	is a positive constant such that Σ_δ is given by <code>sigdelp</code> $\times I_{d(k-1)}$
sigbetap	is a positive constant such that Σ_β is given by <code>sigbetap</code> $\times I_p$
epm	is the $n \times g$ matrix, $epm = \begin{bmatrix} \kappa_{11} & \dots & \kappa_{1g} \\ \vdots & \ddots & \vdots \\ \kappa_{n1} & \dots & \kappa_{ng} \end{bmatrix}$

The output of `greedy`, `greedyXW`, and `greedyXWV` are two components `dur` and `fitpre` which give the duration of fitting the mixture model and the variational approximation respectively. For instance, if we let

```
mixturemodel <- greedy(y,X,W,V,U,vni,epm,M,IGa,IGb,sigdelq,sigbetap).
```

Then `mixturemodel$dur` gives the duration and `mixturemodel$fitpre` gives the variational approximation. We describe the notation used in the code for the components of the variational approximation `mixturemodel$fitpre` below:

<code>mubetaq</code>	$p \times k$ matrix such that $\mu_{\beta_j}^q$ is given by <code>mubetaq[,j]</code>
<code>sigbetaq</code>	array of dimension $k \times p \times p$ such that $\Sigma_{\beta_j}^q$ is given by <code>sigbetaq[j,,]</code>
<code>muaq</code>	$n \times s_1$ matrix such that $\mu_{a_i}^q$ is given by <code>muaq[i,]</code>
<code>sigaq</code>	array of dimension $n \times s_1 \times s_1$ such that $\Sigma_{a_i}^q$ is given by <code>sigaq[i,,]</code>
<code>mubq</code>	$s_2 \times k$ matrix such that $\mu_{b_j}^q$ is given by <code>mubq[,j]</code>
<code>sigbq</code>	array of dimension $k \times s_2 \times s_2$ such that $\Sigma_{b_j}^q$ is given by <code>sigbq[j,,]</code>
<code>alpaq</code>	$k \times 1$ vector such that $\alpha_{a_j}^q$ is given by <code>alpaq[j]</code>
<code>lamaq</code>	$k \times 1$ vector such that $\lambda_{a_j}^q$ is given by <code>lamaq[j]</code>
<code>alpbq</code>	$k \times 1$ vector such that $\alpha_{b_j}^q$ is given by <code>alpbq[j]</code>
<code>lambq</code>	$k \times 1$ vector such that $\lambda_{b_j}^q$ is given by <code>lambq[j]</code>
<code>alpq</code>	$g \times k$ matrix such that α_{jl}^q is given by <code>alpq[l,j]</code>
<code>lamq</code>	$g \times k$ matrix such that λ_{jl}^q is given by <code>lamq[l,j]</code>
<code>qp</code>	$n \times k$ matrix such that q_{ij} is given by <code>qp[i,j]</code>
<code>mudelq</code>	$d(k-1) \times 1$ vector, μ_{δ}^q
<code>sigdelq</code>	$d(k-1) \times d(k-1)$ matrix, Σ_{δ}^q
<code>lb</code>	variational lower bound \mathcal{L}
<code>lbadj</code>	estimated log marginal likelihood

The components of `mixturemodel$fitpre` may be extracted for instance by letting

```
VAFit <- mixturemodel$fitpre
```

and using `VAFit$lb` to obtain the variational lower bound and `VAFit$mudelq` to obtain μ_{δ}^q .

Optional merge steps may be performed using the functions

- `MLMMmerge(y,X,W,V,U,IGa,IGb,sigbetap,sigdelq,vni,epm,tol,fit,m1,m2,type)`
if there is no hierarchical centering
- `MLMPmerge(y,X,W,V,U,IGa,IGb,sigbetap,sigdelq,vni,epm,tol,fit,m1,m2,type)`
for the case of partial centering
- `MLMFmerge(y,X,W,V,U,IGa,IGb,sigbetap,sigdelq,vni,epm,tol,fit,m1,m2,type)`
for the case of full centering

where the arguments y , X , W , V , U , IGa , IGb , IGb , `sigbetap`, `sigdelp`, `vni`, `epm` are as described previously and

- `tol` is the tolerance. We set `tol` as 10^{-5} in the paper which implies that the variational algorithm is terminated when the relative increase in the lower bound is less than 10^{-5} .
- `fit` refers to the variational approximation of the mixture model which contains clusters that we want to merge.
- `m1`, `m2` refers to the clusters of the mixture model that we want to merge. Let `m1` be the larger cluster.
- `type` if `type`='single', only variational parameters of the component arising from the two clusters being merged are updated. If `type`='all', then variational parameters of all components are updated. This option will take more computation time but is more likely to result in an increase in the estimated log marginal likelihood after merging.

Examples on application of the VGA can be found in 'Examples.R'. The first example is on application of VGA using Algorithm 1 (no centering) to the time course data (Spellman *et al.*, 1998) in Section 7.1. The files required are 'ORF_DATA.txt' and 'CDC_DATA.txt' which may be downloaded from <http://www.molbiolcell.org/content/9/12/3273/suppl/DC1>.

The second example is on application of VGA using Algorithm 2 (partial centering) to the completely synthetic data set (Yeung *et al.*, 2003) in Section 7.2. The file required is 'syn_sine_5_mult1' and may be downloaded from <http://expression.washington.edu/publications/kayee/yeunggb2003/> under the section on '4 repeated measurements (400 genes, 20 experiments): Low noise data'.

The third example is on application of VGA using Algorithm 2 (partial centering) to the yeast galactose data (Ideker *et al.*, 2001) discussed in the Supplementary materials. The file required is 'gal205.txt' and may be downloaded from <http://expression.washington.edu/publications/kayee/yeunggb2003/> under the section on 'Log ratio data with repeated measurements (with missing data, tab-delimited text file)'.

The last example is on application of VGA using Algorithm 3 (full centering) to the water temperature data in Section 7.3. The file required is 'temperature_data' provided by the Singapore Delft Water Alliance and is available as part of the supplementary materials.

References

- Ideker, T., Thorsson, V., Ranish, J.A., Christmas, R., Buhler, J., Eng, J.K., Bumgarner, R., Goodlett, D.R., Aebersold, R. and Hood, L. (2001). Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292, 929–934.
- Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6, 461–464.

- Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9, 3273–3297.
- Yeung, K.Y., Medvedovic, M. and Bumgarner, R.E. (2003). Clustering gene-expression data with repeated measurements. *Genome Biology*, 4, Article R34.