Soft-k-means example

- **Compile Stan code**

In[8]:=
```
(* Linux *)
SetDirectory["~/GitHub/MathematicaStan/Examples/Cluster/"];

(* Windows *)
(* SetDirectory["C:\\Users\\USER_NAME\\Documents\\Mathematica\\STAN\\Examples\\Cluster"]; *)

Needs["CmdStan`"];
StanCompile["soft-k-means.stan"] (* CAVEAT: takes some time *)
```

Out[10]=
```
make: '/IS006139/home/pix/GitHub/MathematicaStan/Examples/Cluster/soft-k-means' is up to date.
```

Run generated executable

In[11]:=
```
StanRunVariational["soft-k-means"]
```

Out[11]=
```
method = variational
  variational
    algorithm = meanfield (Default)
      meanfield
    iter = 10000 (Default)
    grad_samples = 1 (Default)
    elbo_samples = 100 (Default)
    eta = 1 (Default)
    adapt
      engaged = 1 (Default)
      iter = 50 (Default)
    tol_rel_obj = 0.01 (Default)
    eval_elbo = 100 (Default)
    output_samples = 1000 (Default)
id = 0 (Default)
data
  file = /IS006139/home/pix/GitHub/MathematicaStan/Examples/Cluster/soft-k-means.data.R
init = 2 (Default)
random
  seed = 3382725612
output
  file = /IS006139/home/pix/GitHub/MathematicaStan/Examples/Cluster/output.csv
  diagnostic_file =  (Default)
  refresh = 100 (Default)


This is Automatic Differentiation Variational Inference.

(EXPERIMENTAL ALGORITHM: expect frequent updates to the procedure.)

Gradient evaluation took 0.000283 seconds
1000 iterations under these settings should take 0.283 seconds.
Adjust your expectations accordingly!

Begin eta adaptation.
Iteration:   1 / 250 [  0%]  (Adaptation)
Iteration:  50 / 250 [ 20%]  (Adaptation)
Iteration: 100 / 250 [ 40%]  (Adaptation)
Iteration: 150 / 250 [ 60%]  (Adaptation)
Iteration: 200 / 250 [ 80%]  (Adaptation)
Success! Found best value [eta = 1] earlier than expected.

Begin stochastic gradient ascent.
  iter       ELBO   delta_ELBO_mean   delta_ELBO_med   notes
   100     -8e+02             1.000            1.000
   200     -8e+02             0.500            1.000
   300     -8e+02             0.334            0.001   MEDIAN ELBO CONVERGED

Drawing a sample of size 1000 from the approximate posterior...
COMPLETED.
```

Import data and variable manipulations

In[12]:=
```
output=StanImport["output.csv"];
```

- **Print header data (20 first variables)**

In[13]:=
```
Take[StanImportHeader[output],20]
```

Out[13]=
```
{{lp__, 1}, {mu.1.1, 2}, {mu.2.1, 3}, {mu.3.1, 4}, {mu.4.1, 5}, {mu.5.1, 6}, {mu.1.2, 7},
 {mu.2.2, 8}, {mu.3.2, 9}, {mu.4.2, 10}, {mu.5.2, 11}, {mu.1.3, 12}, {mu.2.3, 13}, {mu.3.3, 14},
 {mu.4.3, 15}, {mu.5.3, 16}, {mu.1.4, 17}, {mu.2.4, 18}, {mu.3.4, 19}, {mu.4.4, 20}}
```

- **Extract mu for sample 6**

In[14]:=
```
StanVariable["mu", output, 6] // MatrixForm
```

Out[14]//MatrixForm=

$$\begin{pmatrix} -0.633139 & -0.118015 & 1.92222 & -1.58575 & -0.0877965 & -0.495169 & -0.131401 & 0.594322 \\ 1.16202 & 0.251661 & 0.308956 & 1.08092 & 1.98735 & 0.481729 & 1.28065 & 0.106608 \\ 1.97557 & 1.33017 & 1.02126 & 0.786128 & -0.785162 & -0.702112 & -0.341734 & 0.479363 \\ -0.131244 & 0.169858 & -0.46138 & -0.0818914 & -0.246276 & -2.27959 & 0.501451 & 0.458788 \\ -0.825115 & 0.0504496 & 0.477853 & -0.582768 & 1.33536 & 0.138424 & 0.0823158 & 0.642028 \end{pmatrix}$$

In[15]:=
```
StanVariable["mu.2.3",output,6]
```

Out[15]= {0.308956}

- **Extract the whole column of sample for mu.2.3 (only print the first 10)**

In[16]:=
```
Take[StanVariableColumn["mu.2.3", output], 10] // MatrixForm
```

Out[16]//MatrixForm=

$$\begin{pmatrix} 0.587841 \\ 0.511622 \\ 0.483623 \\ 0.450044 \\ 0.895536 \\ 0.308956 \\ 0.603064 \\ 0.370355 \\ 0.628372 \\ 0.643755 \end{pmatrix}$$

- **Compute mean and standard deviation for the mu variable**

In[17]:=
```
StanVariableFunc["mu", output, Mean] // MatrixForm
StanVariableFunc["mu", output, StandardDeviation] // MatrixForm
```

Out[17]//MatrixForm=

$$\begin{pmatrix} -0.632107 & -0.0369934 & 2.10781 & -1.36347 & -0.205826 & -0.41219 & -0.135744 & 0.55432 \\ 1.189 & 0.0688103 & 0.597086 & 0.994098 & 1.74106 & 0.333556 & 0.988289 & -0.0137025 \\ 1.40013 & 1.65572 & 0.96985 & 0.628772 & -0.613356 & -1.35198 & -0.354806 & 0.294138 \\ 0.0734767 & 0.1476 & -0.423422 & 0.185093 & -0.569207 & -2.35743 & 0.730132 & -0.00367877 \\ -0.547612 & 0.385826 & 0.491658 & -0.342182 & 1.55178 & -0.0280413 & 0.231809 & 0.3384 \end{pmatrix}$$

Out[18]//MatrixForm=

$$\begin{pmatrix} 0.185271 & 0.17174 & 0.185272 & 0.141742 & 0.165811 & 0.162659 & 0.187898 & 0.141319 \\ 0.215834 & 0.219047 & 0.209776 & 0.182311 & 0.259483 & 0.184783 & 0.210824 & 0.21377 \\ 0.317949 & 0.258503 & 0.265444 & 0.275148 & 0.314137 & 0.347332 & 0.256 & 0.252446 \\ 0.200921 & 0.190878 & 0.185952 & 0.194381 & 0.196106 & 0.205363 & 0.227025 & 0.174352 \\ 0.192755 & 0.276548 & 0.316786 & 0.307607 & 0.352852 & 0.281774 & 0.317026 & 0.241737 \end{pmatrix}$$