Soft-k-means example

- **Compile Stan code**

In[19]:=
```
(* 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[21]=
```
make: '/IS006139/home/pix/GitHub/MathematicaStan/Examples/Cluster/soft-k-means' is up to date.
```

Run generated executable

In[4]:=
```
StanRunVariational["soft-k-means"]
```

Out[4]=
```
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 = 3919302228
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 0.000279 seconds
1000 iterations under these settings should take 0.279 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             0.501            1.000
   300     -8e+02             0.335            0.003   MEDIAN ELBO CONVERGED

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

Import data and variable manipulations

In[5]:=
```
output=StanImport["output.csv"];
```

Print header data (20 first variables)

In[6]:=
```
Take[StanImportHeader[output],20]
```

Out[6]=
```
{{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}}
```

Print data as a dense matrix

Here restricted to print only the first 10x10 submatrix

In[7]:=
```
Take[StanImportData[output],10,10]
```

Out[7]=
```
{{0., -0.596362, 1.20387, -0.466482, 1.35331, -0.0913661, 0.0665724, 0.0575724, 0.659211, 1.60823},
 {0., -0.463683, 1.18395, -0.391218, 1.46582, -0.167954, 0.121784, 0.281143, 0.293225, 1.33609},
 {0., -0.508731, 0.969964, -0.374607, 1.22813, -0.382994, 0.0651003, 0.262559, 0.614845, 1.58483},
 {0., -0.708209, 1.10826, -0.764628, 1.42732, 0.192764, 0.266772, 0.0107016, 1.09144, 1.58153},
 {0., -0.780292, 1.27679, -0.460673, 1.10943, 0.00896912, -0.148645, 0.00212006, 0.472678, 1.26102},
 {0., -0.523696, 1.39001, -0.684258, 1.4678, 0.181688, -0.342918, 0.0650176, 0.571362, 1.52378},
 {0., -0.800905, 1.28875, -0.679895, 1.48111, -0.393382, -0.0679261, -0.0730522, 0.693933, 1.75825},
 {0., -0.426127, 0.932423, -0.436705, 0.865112, -0.266308, -0.369898, -0.070338, 0.532156, 1.42951},
 {0., -0.558727, 1.58077, -0.58597, 1.56047, -0.520021, 0.0575043, -0.119397, 0.693666, 1.8539},
 {0., -0.469435, 1.59463, -0.316618, 1.85334, -0.129154, 0.175447, 0.0438011, 0.81903, 1.35904}}
```

Print solver output

In[8]:=
```
StanImportComment[output]
```

Out[8]=
```
# stan_version_major = 2
# stan_version_minor = 11
# stan_version_patch = 0
# model = soft_k_means_model
# 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 = 3919302228
# output
#   file = /IS006139/home/pix/GitHub/MathematicaStan/Examples/Cluster/output.csv
#   diagnostic_file =  (Default)
#   refresh = 100 (Default)
# Stepsize adaptation complete.
# eta = 1
```

Extract μ for sample 6

In[9]:=
```
StanVariable["mu",output,6] // MatrixForm
```

Out[9]//MatrixForm=

−0.523696	−0.342918	1.75112	−1.53009	−0.37731	−0.202779	0.00153212	0.698405
1.39001	0.0650176	0.114315	0.510581	1.62991	0.199937	1.08333	0.147248
−0.684258	0.571362	0.2377	−0.559675	1.21017	−0.279591	−0.29057	0.365839
1.4678	1.52378	0.613555	0.747827	−0.645775	−1.32907	0.118526	0.219675
0.181688	0.195762	−0.596392	0.441758	−0.353317	−2.41884	0.882771	−0.0344794

In[10]:=
```
StanVariable["mu.2.3",output,6]
```

Out[10]= {0.114315}

Compute μ Variance and Mean

In[11]:=
```
StanVariableFunc["mu",output,Variance] // MatrixForm
StanVariableFunc["mu",output,Mean] // MatrixForm
```

Out[11]//MatrixForm=

0.0268518	0.053186	0.0361522	0.0451887	0.0295765	0.032995	0.0340826	0.0265834
0.10542	0.049589	0.0413606	0.035541	0.037861	0.0405695	0.0525858	0.0395567
0.0737501	0.0759013	0.12784	0.0763452	0.0787031	0.0684789	0.0805156	0.0585508
0.0849128	0.0687241	0.049207	0.047619	0.0438936	0.0776381	0.0574155	0.0928793
0.0687237	0.0358052	0.040172	0.0588392	0.0448996	0.0418819	0.0398141	0.0432061

Out[12]//MatrixForm=

−0.595748	0.0708949	2.00972	−1.42061	−0.265239	−0.229582	−0.0177429	0.68803
1.20199	0.0519999	0.569316	0.898124	1.6881	0.230242	1.09694	0.0517219
−0.468303	0.660436	0.413128	−0.475021	1.45084	−0.126665	0.0291967	0.360241
1.33749	1.61756	0.900774	0.613851	−0.688265	−1.30466	−0.145496	0.317668
−0.098543	0.0766514	−0.440867	0.279155	−0.634081	−2.47228	0.691151	−0.0196336

Advanced output manipulation

Find column indices

In[13]:=
```
StanFindVariableColumn["mu", output]
StanFindVariableColumn["mu.2", output]
StanFindVariableColumn["mu.2.3", output]
```

Out[13]= {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41}

Out[14]= {3, 8, 13, 18, 23, 28, 33, 38}

Out[15]= {13}

Find variable indices

In[16]:=
```
StanFindVariableIndex["mu", output]
StanFindVariableIndex["mu.2", output]
StanFindVariableIndex["mu.2.3", output]
```

Out[16]= {{1, 1}, {2, 1}, {3, 1}, {4, 1}, {5, 1}, {1, 2}, {2, 2}, {3, 2}, {4, 2},
{5, 2}, {1, 3}, {2, 3}, {3, 3}, {4, 3}, {5, 3}, {1, 4}, {2, 4}, {3, 4}, {4, 4},
{5, 4}, {1, 5}, {2, 5}, {3, 5}, {4, 5}, {5, 5}, {1, 6}, {2, 6}, {3, 6}, {4, 6},
{5, 6}, {1, 7}, {2, 7}, {3, 7}, {4, 7}, {5, 7}, {1, 8}, {2, 8}, {3, 8}, {4, 8}, {5, 8}}

Out[17]= {{2, 1}, {2, 2}, {2, 3}, {2, 4}, {2, 5}, {2, 6}, {2, 7}, {2, 8}}

Out[18]= {{2, 3}}

Extract associated columns

Here I only take the first 10 rows (and even the first 16 columns of the full mu matrix)

In[69]:=
```
Take[StanVariableColumn["mu", output], 10, 16] // MatrixForm
Take[StanVariableColumn["mu.2", output], 10] // MatrixForm
Take[StanVariableColumn["mu.2.3", output], 10] // MatrixForm
```

Out[69]//MatrixForm=

1.27646	1.31632	0.0151894	−0.572325	−0.554552	1.79741	0.256787	0.0318775	0.314616	0.00099
1.17222	1.33609	0.166674	−0.604865	−0.209279	1.35317	0.248634	0.240779	0.431254	−0.0257
1.4301	0.971868	0.0346061	−0.274483	−0.511333	2.04893	−0.0375499	0.241264	0.371024	0.1352
1.27587	1.09322	0.356328	0.0464723	−0.50384	1.56323	0.250389	0.183071	0.555089	0.0834
1.3531	1.61101	0.196676	−0.93089	−0.143187	2.18611	0.0704879	−0.0611038	0.32735	−0.210
1.0956	1.17351	0.0707404	−0.489752	−0.279649	1.80807	0.183232	0.140403	0.262518	−0.131
1.09412	1.08351	0.296302	−0.693432	−0.318768	1.60794	0.137077	0.308576	0.119771	0.00773
1.30726	1.2054	−0.284553	−0.259875	−0.318144	1.72322	0.194123	−0.148456	0.816033	0.1493
1.17144	1.23669	−0.0847276	−1.27247	−0.616423	1.14577	−0.0251561	0.129403	0.576241	−0.128
1.46217	1.60942	−0.113072	−0.317871	−0.53687	1.96188	0.252162	0.275657	0.174242	−0.0926

Out[70]//MatrixForm=

1.31632	0.256787	0.446693	0.983715	1.73196	0.294308	0.877217	0.00325735
1.33609	0.248634	0.495013	1.13979	1.71828	0.00796475	1.01956	0.335615
0.971868	−0.0375499	0.302779	1.15344	1.0656	0.271063	0.854082	−0.171532
1.09322	0.250389	0.536753	1.55645	2.01183	0.77971	1.09258	−0.0624234
1.61101	0.0704879	0.654006	1.00371	1.58953	0.107967	0.726299	−0.154444
1.17351	0.183232	0.496767	1.46877	1.69367	0.0217598	0.861771	0.0110387
1.08351	0.137077	0.567158	1.03385	1.26696	0.10625	1.09166	0.254345
1.2054	0.194123	0.150004	0.762519	1.40222	0.139547	0.884385	0.343576
1.23669	−0.0251561	0.554002	1.19911	1.63321	−0.0278946	0.78595	0.358755
1.60942	0.252162	0.359385	0.88132	1.92574	0.120109	0.90541	−0.262331

Out[71]//MatrixForm=

0.446693
0.495013
0.302779
0.536753
0.654006
0.496767
0.567158
0.150004
0.554002
0.359385