Problem 3

In [2]:

```python
import load_helper

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

import importlib

from torch import tensor
```

In [53]:

```python
fname = '3.daphne'
graph = load_helper.graph_helper(fname)
%cat 3.daphne
```

```
(let [data [1.1 2.1 2.0 1.9 0.0 -0.1 -0.05]
      likes (foreach 3 []
                     (let [mu (sample (normal 0.0 10.0))
                           sigma (sample (gamma 1.0 1.0))]
                       (normal mu sigma)))
      pi (sample (dirichlet [1.0 1.0 1.0]))
      z-prior (discrete pi)
      z (foreach 7 [y data]
          (let [z (sample z-prior)
                _ (observe (get likes z) y)]
            z))]
  (= (first z) (second z)))
```

In [16]:

```python
import bbvi
importlib.reload(bbvi)

from bbvi import graph_bbvi_algo12
```

In [34]:

```python
%%time
T=1000
L=100
lr=0.05
r, logW, sigma = bbvi.graph_bbvi_algo12(graph,T=T,L=L,lr=lr,
                                        do_log=False)
```

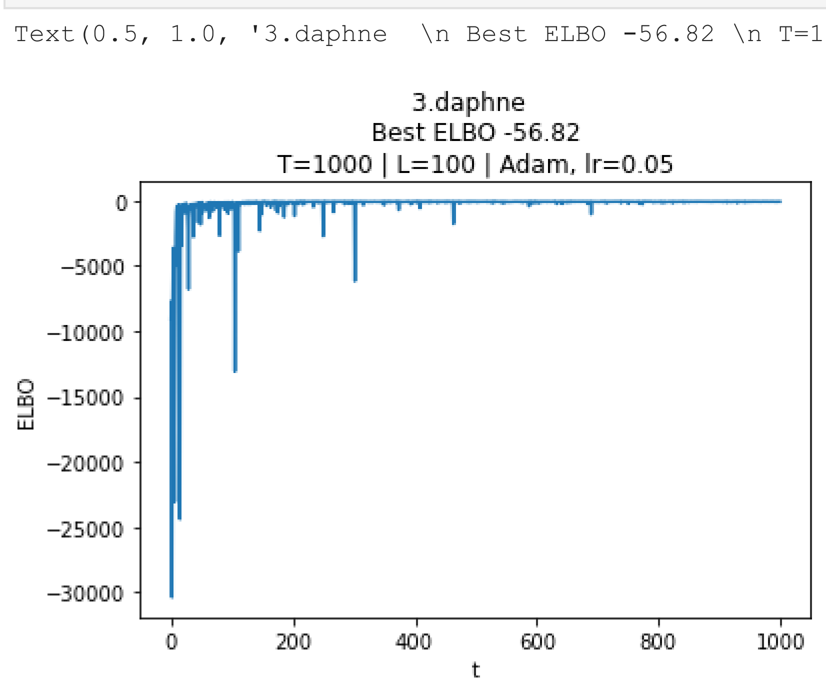
```
t=0, Q after step={'sample3': Gamma(concentration: 1.0499999523162842, rate: 0.54132479429245), 'sample1': Gamm
a(concentration: 1.0499999523162842, rate: 0.54132479429245), 'sample2': Normal(loc: -0.05000000074505806, scal
e: 9.999954223632812), 'sample4': Normal(loc: -0.04999999701976776, scale: 9.999954223632812), 'sample6': Diric
hlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size([3])),
'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(probs: torch.
Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Size([3])),
'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs: torch.Si
ze([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 's
ample0': Normal(loc: 0.05000000074505806, scale: 9.999954223632812), 'sample5': Gamma(concentration: 0.94999998
8079071, rate: 0.54132479429245)}
t=100, Q after step={'sample3': Gamma(concentration: 1.7900134325027466, rate: 0.3037601709365845), 'sample1':
Gamma(concentration: 1.8541914224624634, rate: 0.27189576625823975), 'sample2': Normal(loc: 0.1060582771897316,
scale: 10.16331672668457), 'sample4': Normal(loc: 0.2834552824497223, scale: 10.029430389404297), 'sample6': Di
richlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(prob
s: torch.Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Siz
e([3])), 'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs:
torch.Size([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample0': Normal(loc: 0.4662581980228424, scale: 9.678585052490234), 'sample5': Gamma(concentration:
1.4942097663879395, rate: 0.3359958529472351)}
t=200, Q after step={'sample3': Gamma(concentration: 1.8218340873718262, rate: 0.288897842168808), 'sample1': G
amma(concentration: 2.0777390003204346, rate: 0.21965110301971436), 'sample2': Normal(loc: 0.428939551115036, s
cale: 9.960811614990234), 'sample4': Normal(loc: 0.5669648051261902, scale: 9.839775085449219), 'sample6': Diri
chlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(prob
s: torch.Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Siz
e([3])), 'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs:
torch.Size([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample0': Normal(loc: 0.9046857953071594, scale: 9.574071884155273), 'sample5': Gamma(concentration:
1.7429133653640747, rate: 0.27125629782676697)}
t=300, Q after step={'sample3': Gamma(concentration: 1.8476452827453613, rate: 0.28053200244903564), 'sample1':
Gamma(concentration: 2.1341969966888428, rate: 0.20382164418697357), 'sample2': Normal(loc: 0.4442814290523529,
scale: 9.737642288208008), 'sample4': Normal(loc: 0.38011208176612854, scale: 9.466002464294434), 'sample6': Di
richlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(prob
s: torch.Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Siz
e([3])), 'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs:
torch.Size([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample0': Normal(loc: 0.8249368667602539, scale: 9.532282829284668), 'sample5': Gamma(concentration:
1.8130440711975098, rate: 0.2528989017009735)}
t=400, Q after step={'sample3': Gamma(concentration: 1.8373630046844482, rate: 0.29939261078834534), 'sample1':
Gamma(concentration: 2.2015764713287354, rate: 0.17110460996627808), 'sample2': Normal(loc: 0.47037473320961, s
cale: 9.841106414794922), 'sample4': Normal(loc: 0.22893062233924866, scale: 9.439980506896973), 'sample6': Dir
ichlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(prob
s: torch.Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Siz
e([3])), 'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs:
torch.Size([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample0': Normal(loc: 1.2280818223953247, scale: 9.21605110168457), 'sample5': Gamma(concentration: 1.
9015353918075562, rate: 0.19762399792671204)}
t=500, Q after step={'sample3': Gamma(concentration: 1.8658350706100464, rate: 0.2749086618423462), 'sample1':
Gamma(concentration: 2.231902837753296, rate: 0.15848882496356964), 'sample2': Normal(loc: 0.47287917137145996,
scale: 9.806499481201172), 'sample4': Normal(loc: 0.32633745670318604, scale: 9.028315544128418), 'sample6': Di
richlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(prob
s: torch.Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Siz
e([3])), 'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs:
torch.Size([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample0': Normal(loc: 1.039696216583252, scale: 9.088177680969238), 'sample5': Gamma(concentration: 1.
9733400344848633, rate: 0.17851786315441132)}
t=600, Q after step={'sample3': Gamma(concentration: 1.8757519721984863, rate: 0.26896676421165466), 'sample1':
Gamma(concentration: 2.243215322494507, rate: 0.15821991860866547), 'sample2': Normal(loc: 0.4916940927505493,
scale: 9.686244010925293), 'sample4': Normal(loc: 0.3182417154312134, scale: 8.951693534851074), 'sample6': Dir
ichlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(prob
s: torch.Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Siz
e([3])), 'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs:
torch.Size([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample0': Normal(loc: 1.0337268114089966, scale: 9.07003116607666), 'sample5': Gamma(concentration: 1.
9913407564163208, rate: 0.17566214501857758)}
t=700, Q after step={'sample3': Gamma(concentration: 1.8960187435150146, rate: 0.2543301284313202), 'sample1':
Gamma(concentration: 2.267752170562744, rate: 0.15137308835983276), 'sample2': Normal(loc: 0.381063848733902, s
cale: 9.35001277923584), 'sample4': Normal(loc: 0.40551134943962097, scale: 8.892863273620605), 'sample6': Diri
chlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(prob
s: torch.Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Siz
e([3])), 'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs:
torch.Size([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample0': Normal(loc: 0.9871072769165039, scale: 9.003787994384766), 'sample5': Gamma(concentration:
2.0286083221435547, rate: 0.16498127579689026)}
t=800, Q after step={'sample3': Gamma(concentration: 1.9163165092468262, rate: 0.2426215559244156), 'sample1':
Gamma(concentration: 2.2835867404937744, rate: 0.1482381969690323), 'sample2': Normal(loc: 0.3498328626155853,
scale: 9.097940444946289), 'sample4': Normal(loc: 0.4434855282306671, scale: 8.850457191467285), 'sample6': Dir
ichlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(prob
s: torch.Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Siz
e([3])), 'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs:
torch.Size([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample0': Normal(loc: 1.0016863346099854, scale: 8.960394859313965), 'sample5': Gamma(concentration:
2.0418074131011963, rate: 0.16577433049678802)}
t=900, Q after step={'sample3': Gamma(concentration: 1.9285883903503418, rate: 0.23700419068336487), 'sample1':
Gamma(concentration: 2.2909774780273438, rate: 0.15304233133792877), 'sample2': Normal(loc: 0.3641284108161926,
scale: 9.029130935668945), 'sample4': Normal(loc: 0.443512886762619, scale: 8.797557830810547), 'sample6': Diri
chlet(concentration: torch.Size([3])), 'sample11': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample13': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample19': Categorical(prob
s: torch.Size([3]), logits: torch.Size([3])), 'sample15': Categorical(probs: torch.Size([3]), logits: torch.Siz
e([3])), 'sample9': Categorical(probs: torch.Size([3]), logits: torch.Size([3])), 'sample7': Categorical(probs:
torch.Size([3]), logits: torch.Size([3])), 'sample17': Categorical(probs: torch.Size([3]), logits: torch.Size
([3])), 'sample0': Normal(loc: 1.0233283042907715, scale: 8.92258358001709), 'sample5': Gamma(concentration: 2.
0463082790374756, rate: 0.17054669559001923)}
CPU times: user 48min 53s, sys: 6.19 s, total: 48min 59s
Wall time: 2h 13min 5s
```

In [36]:

```python
elbo = logW.mean(1)
pd.Series(elbo).plot()
plt.xlabel('t')
plt.ylabel('ELBO')
plt.title('{}  \n Best ELBO {:1.2f} \n T={} | L={} | Adam, lr={} '.format(fname,elbo.max(),T,L,lr))
```

Out[36]:

```
Text(0.5, 1.0, '3.daphne  \n Best ELBO -56.82 \n T=1000 | L=100 | Adam, lr=0.05 ')
```



In [37]:

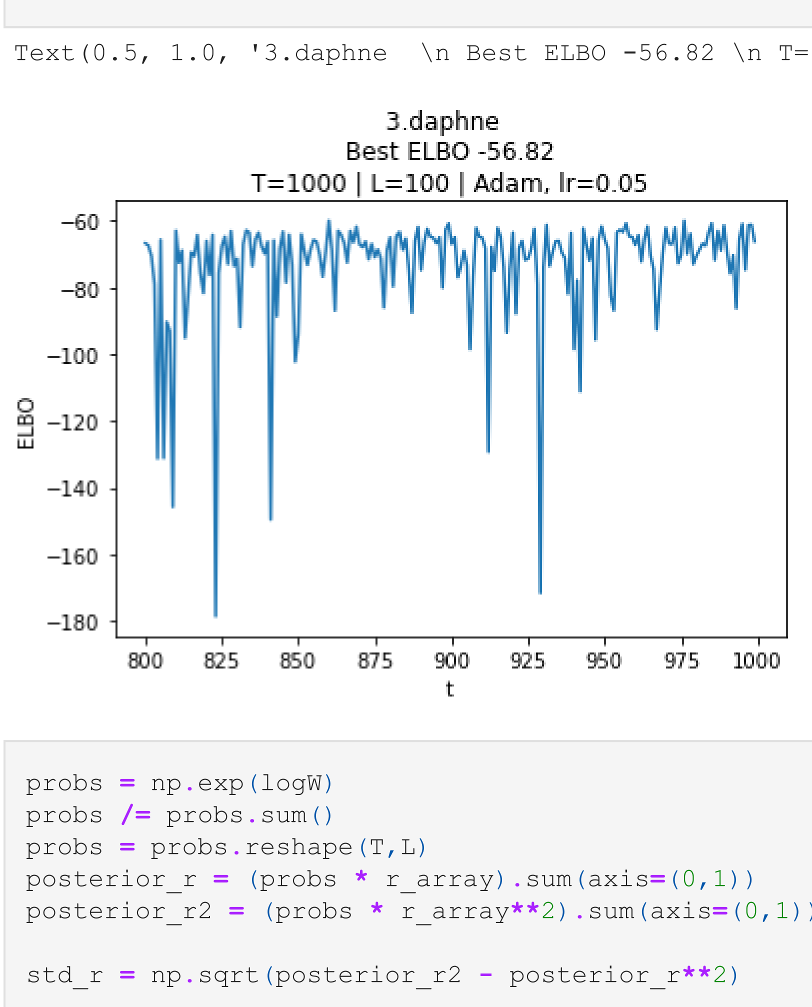
```python
r_array = np.array(r)
```

In [39]:

```python
sr = pd.Series(elbo[-200:])
sr.index = np.arange(elbo.size-200,elbo.size)
sr.plot()
plt.xlabel('t')
plt.ylabel('ELBO')
plt.title('{}  \n Best ELBO {:1.2f} \n T={} | L={} | Adam, lr={} '.format(fname,elbo.max(),T,L,lr))
```

Out[39]:

```
Text(0.5, 1.0, '3.daphne  \n Best ELBO -56.82 \n T=1000 | L=100 | Adam, lr=0.05 ')
```



In [42]:

```python
probs = np.exp(logW)
probs /= probs.sum()
probs = probs.reshape(T,L)
posterior_r = (probs * r_array).sum(axis=(0,1))
posterior_r2 = (probs * r_array**2).sum(axis=(0,1))

std_r = np.sqrt(posterior_r2 - posterior_r**2)
```

In [51]:

```python
print('{} expected posterior z1==z2 {:1.3f} | std posterior z1==z2 {:1.3f}'.format(fname,posterior_r,std_r))
```

```
3.daphne expected posterior z1==z2 0.049 | std posterior z1==z2 0.215
```

Also write a paragraph or two about the mode-seeking behavior of VI on models like this that have internal symmetries. Note what the symmetries are and why they might cause problems with optimization.

This problem is a Gaussian mixture model. As discussed in office hours, the choice of KL(p||q) instead of KL(q||q) makes us snap to one mode, in this multimodal problem. See the inclusive divergence and exclusive divergence remarks on Stefano Ermon's Probabilistic Graphical Models Winter class (Stanford 2020-21). This GMM problem is multimodal, because there is no meaning to the cluster labels attached to the data points, they are just arbitrary labels. Stochastically, if some datapoints are relabelled, the cluster can shift in the sense that, the labels can swap between clusters. When this is happening in the algorithm, we are "jumping out" of a local minima, and "jumping into" another.