Branch: master pymc3 / docs / source / notebooks / gaussian_mixture_model.ipynb Find file Copy path

twiecki Update more NBs. Remove Dawid-Skene. a3622d3 on Apr 17

1 contributor

333 lines (332 sloc) 653 KB Raw Blame History

Gaussian Mixture Model

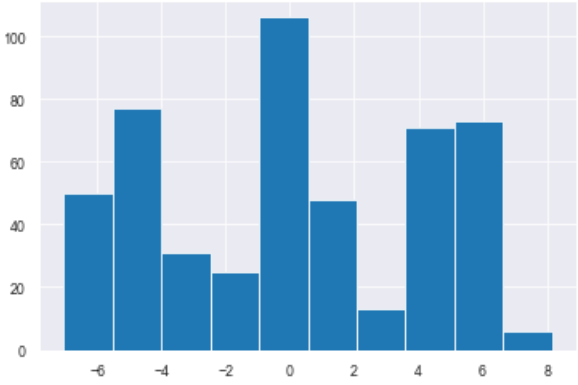
Original NB by Abe Flaxman, modified by Thomas Wiecki

In [1]:
```
!date
import numpy as np, pandas as pd, matplotlib.pyplot as plt, seaborn as sns
%matplotlib inline
sns.set_context('paper')
sns.set_style('darkgrid')
```

```
Tue Apr 16 17:29:15 CEST 2019
```

In [2]:
```
import pymc3 as pm, theano.tensor as tt
```

In [3]:
```
# simulate data from a known mixture distribution
np.random.seed(12345) # set random seed for reproducibility

k = 3
ndata = 500
spread = 5
centers = np.array([-spread, 0, spread])

# simulate data from mixture distribution
v = np.random.randint(0, k, ndata)
data = centers[v] + np.random.randn(ndata)

plt.hist(data);
```



In [5]:
```
# setup model
model = pm.Model()
with model:
    # cluster sizes
    p = pm.Dirichlet('p', a=np.array([1., 1., 1.]), shape=k)
```

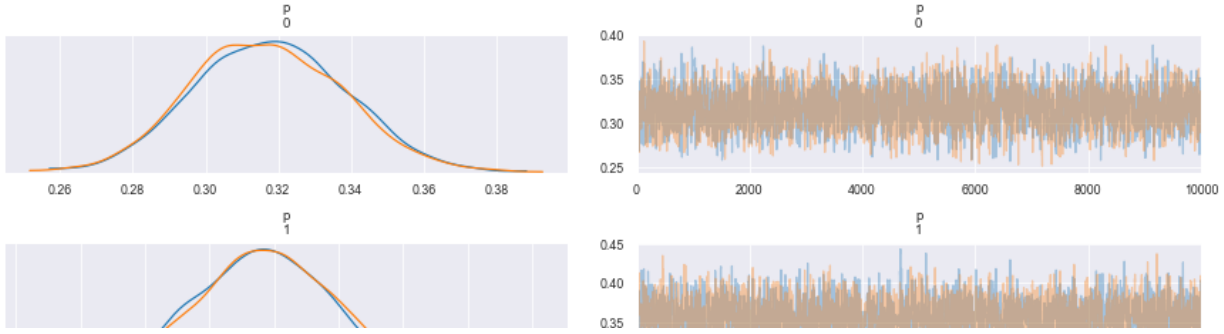
```
    # ensure all clusters have some points
    p_min_potential = pm.Potential('p_min_potential', tt.switch(tt.min(p) < .1, -np.inf, 0))


    # cluster centers
    means = pm.Normal('means', mu=[0, 0, 0], sigma=15, shape=k)
    # break symmetry
    order_means_potential = pm.Potential('order_means_potential',
                                         tt.switch(means[1]-means[0] < 0, -np.inf, 0)
                                         + tt.switch(means[2]-means[1] < 0, -np.inf, 0))

    # measurement error
    sd = pm.Uniform('sd', lower=0, upper=20)

    # latent cluster of each observation
    category = pm.Categorical('category',
                              p=p,
                              shape=ndata)

    # likelihood for each observed value
    points = pm.Normal('obs',
                       mu=means[category],
                       sigma=sd,
                       observed=data)
```

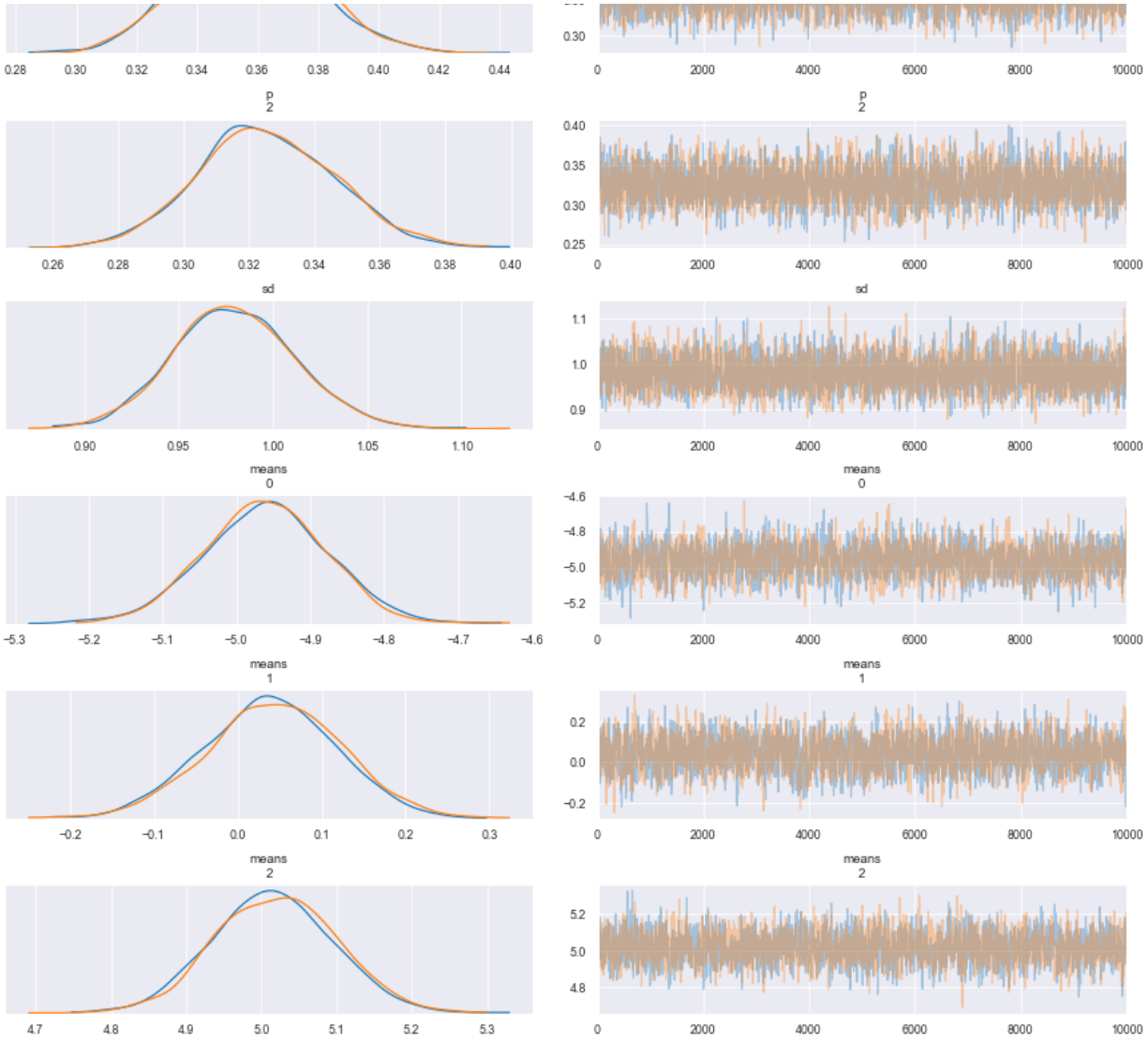
```
INFO (theano.gof.compilelock): Waiting for existing lock by process '70988' (I am process '71002')
INFO (theano.gof.compilelock): To manually release the lock, delete /Users/twiecki/.theano/compiledi
r_Darwin-18.5.0-x86_64-i386-64bit-i386-3.6.7-64/lock_dir
```

In [ ]:
```
# fit model
with model:
    step1 = pm.Metropolis(vars=[p, sd, means])
    step2 = pm.ElemwiseCategorical(vars=[category], values=[0, 1, 2])
    tr = pm.sample(10000, step=[step1, step2], tune=5000)
```

Full trace

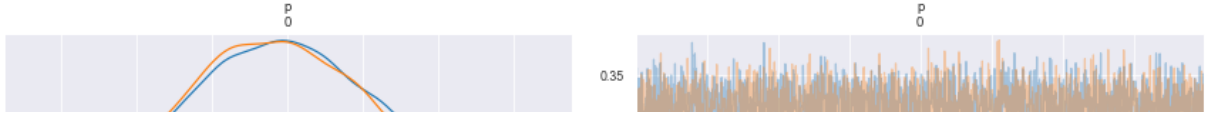
In [7]:
```
pm.traceplot(tr, var_names=['p', 'sd', 'means']);
```

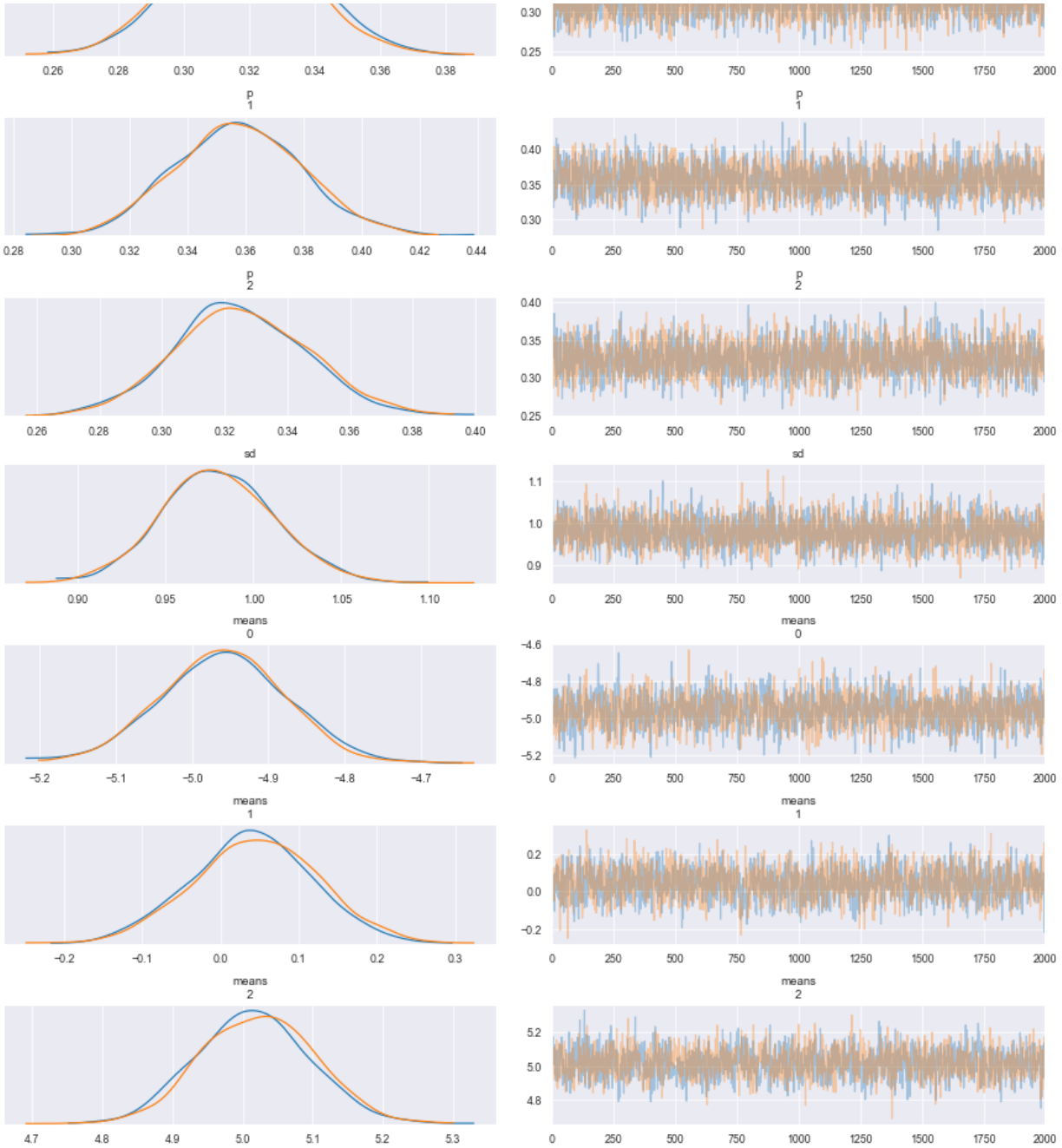




After convergence

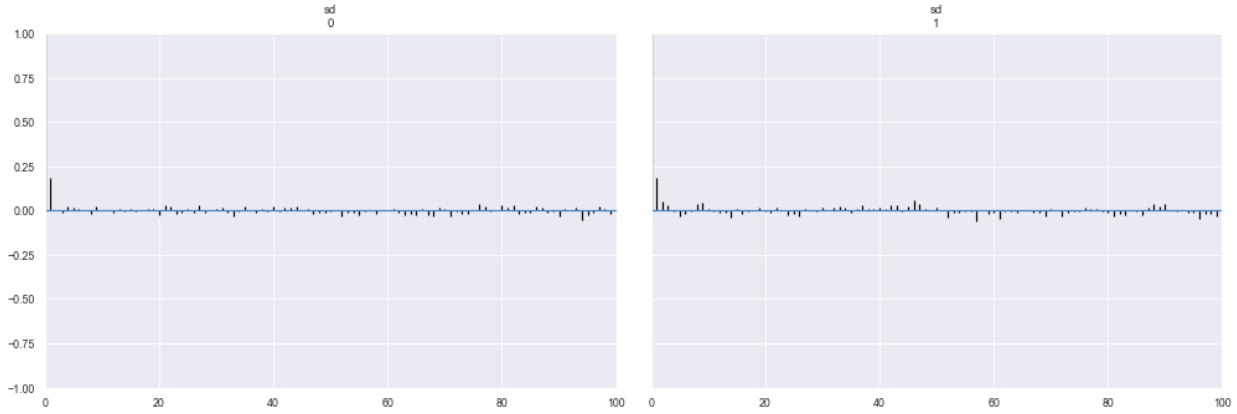
In [8]:
```
# take a look at traceplot for some model parameters
pm.plots.traceplot(tr[::5], var_names=['p', 'sd', 'means']);
```





In [12]:
```
# I prefer autocorrelation plots for serious confirmation of MCMC convergence
pm.autocorrplot(tr[::5], var_names=['sd']);
```

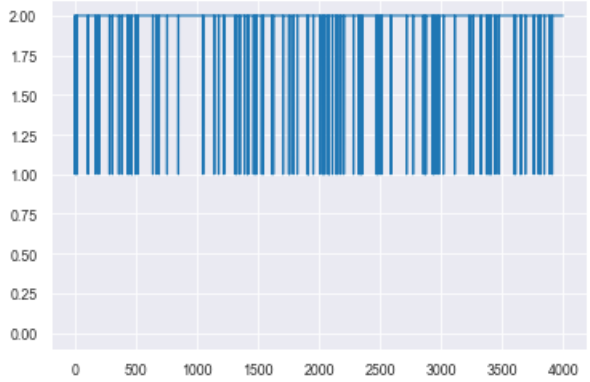
```
/Users/twiecki/anaconda3/lib/python3.6/site-packages/mkl_fft/_numpy_fft.py:1044: FutureWarning: Usin
g a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which wi
ll result either in an error or a different result.
  output = mkl_fft.rfftn_numpy(a, s, axes)
```



Sampling of cluster for individual data point

In [10]:
```
i=0
plt.plot(tr['category'][::5, i], drawstyle='steps-mid')
plt.axis(ymin=-.1, ymax=2.1)
```

Out[10]: (-199.95000000000002, 4198.95, -0.1, 2.1)



In [11]:
```
def cluster_posterior(i=0):
    print('true cluster:', v[i])
    print('  data value:', np.round(data[i],2))
    plt.hist(tr['category'][::5,i], bins=[-.5,.5,1.5,2.5,], rwidth=.9)
    plt.axis(xmin=-.5, xmax=2.5)
    plt.xticks([0,1,2])
```

```
cluster_posterior(i)
```

```
true cluster: 2
  data value: 3.29
```