microsoft / VAEM

Sample code for VAEM: a Deep Generative Model for Heterogeneous Mixed Type Data

View license

2 stars 2 forks

Unstar Watch

Code Issues Pull requests Actions Projects Wiki Security Insigl

main ...

VAEM / Main Notebook.ipynb

v-chma2 Upload VAEM code History

0 contributors

Raw Blame

890 lines (890 sloc) | 153 KB

VAEM: a Deep Generative Model for Heterogeneous Mixed Type Data

VAEM is an extension of variational autoencoders (VAEs) in order to handle such heterogeneous data. It is a deep generative model that is trained in a two stage manner. In the first stage we fit a different VAE independently to each data dimension x_{nd}. We call the resulting D models marginal VAEs. Then, in the second stage, in order to capture the inter-variable dependencies, a new multi-dimensional VAE, called the dependency network, is build on top of the latent representations provided by the first-stage encoders. Finally, if the model is used in down stream tasks such as sequential active information acquisition, we often introduce a third stage, which is to add a new discriminator (preditor) model on top of the VAEM outputs.

Different stages are referred as `1`,`2`, and `3` in `list_stages` in the `.json` files.

Usage

To run the demo, you need to first download the Bank Marketing UCI dataset (https://archive.ics.uci.edu/ml/datasets/Bank+Marketing), and put the csv file under `data/bank`. You will need to preprocess the data into the format according to our example .csv file (which does not contain any real data). This can done by splitting the text into columns using `;` as delimiters. Then, simply run `Main_Notebook.ipynb`. This notebook train/or load a VAEM model on Bank dataset, and demonstrates how to perform sequential active information acquisition (SAIA) and imputation. By default, it trains a new model on Bank dataset. If you would like to load a pre-trained model, by default it will load a pre-trained tensorflow model from `saved_weights/bank/`. Note that in order to perform active information acquisition, an additional third stage training is required. This will add a discriminator (predictor) to the model, which is required for SAIA. The configurations for VAEM can be found in `.json` files in `hyperparameters/bank`, which include:

- "list_stage" : list of stages that you would like the model to be trained. stage 1 = training marginal VAEs, stage 2 = training dependency network, stage 3 = add predictor and improve predictive performance. The default is [1,2].
- "epochs" : number of epochs for training VAEM. If you would like to load a pretrained model rather than training a new one, you can simply set this to zero.
- "latent_dim" : size of latent dimensions of dependency network,
- "p" : upper bound for artificial missingness probability. For example, if set to 0.9, then during each training epoch, the algorithm will randomly choose a probability smaller than 0.9, and randomly drops observations according to this probability. Our suggestion is that if original dataset already contains missing data, you can just set p to 0.
- "iteration" : iterations (number of mini batches) used per epoch. set to -1 to run the full epoch. If your dataset is large, please set to other values such as 10.
- "batch_size" : iterations (number of mini batches) used per epoch. set to -1 to run the full epoch. If your dataset is large, please set to other values such as 10.
- "K" : the dimension of the feature map (h) dimension of PNP encoder.
- "M" : Number of MC samples when perform imputing.
- "repeat" : number of repeats.
- "data_name" : name of the dataset being used. Our default is "bank".
- "output_dir" : Directory where the model is stored. Our default is "./saved_weights/bank/",
- "data_dir" : Directory where the data is stored. Our default is "./data/bank/",
- "list_strategy" : list of strategies for active learning, 0 = random, 1 = single ordering. Default: [1]

Load modules

In [1]:
```
import numpy as np
```

In [1]:
```
import numpy as np
import tensorflow as tf
print(tf.__version__)
from scipy.stats import bernoulli
import os
import random
from random import sample
from sklearn.feature_selection import mutual_info_regression
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import pandas as pd
import sklearn.preprocessing as preprocessing
from sklearn.metrics import mean_squared_error
from sklearn.feature_selection import mutual_info_regression, mutual_info_clas
sif
plt.switch_backend('agg')
tfd = tf.contrib.distributions
import utils.process as process
import json
import utils.params as params
import seaborn as sns; sns.set(style="ticks", color_codes=True)
```

```
/home/paperspace/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: Fu
tureWarning: Conversion of the second argument of issubdtype from `float` to `
np.floating` is deprecated. In future, it will be treated as `np.float64 == n
p.dtype(float).type`.
  from ._conv import register_converters as _register_converters
1.4.1
```

load hyperparameters

In [2]:
```
args = params.Params('./hyperparameters/bank_plot.json')

if not os.path.exists(args.output_dir):
    os.makedirs(args.output_dir)
rs = 42 # random seed
fast_plot = 0
```

Load Bank Data

In [3]:
```
seed = 3000
bank_raw = pd.read_csv("./data/bank/bankmarketing_train.csv")
print(bank_raw.info())
label_column="y"
matrix1 = bank_raw.copy()


process.encode_catrtogrial_column(matrix1, ["job"])
process.encode_catrtogrial_column(matrix1, ["marital"])
process.encode_catrtogrial_column(matrix1, ["education"])
process.encode_catrtogrial_column(matrix1, ["default"])
process.encode_catrtogrial_column(matrix1, ["housing"])
process.encode_catrtogrial_column(matrix1, ["loan"])
process.encode_catrtogrial_column(matrix1, ["contact"])
process.encode_catrtogrial_column(matrix1, ["month"])
process.encode_catrtogrial_column(matrix1, ["day_of_week"])
process.encode_catrtogrial_column(matrix1, ["poutcome"])
process.encode_catrtogrial_column(matrix1, ["y"])

Data = ((matrix1.values).astype(float))[0:,:]
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32950 entries, 0 to 32949
```

```
RangeIndex: 32950 entries, 0 to 32949
Data columns (total 21 columns):
age               32950 non-null int64
job               32950 non-null object
marital           32950 non-null object
education         32950 non-null object
default           32950 non-null object
housing           32950 non-null object
loan              32950 non-null object
contact           32950 non-null object
month             32950 non-null object
day_of_week       32950 non-null object
duration          32950 non-null int64
campaign          32950 non-null int64
pdays             32950 non-null int64
previous          32950 non-null int64
poutcome          32950 non-null object
emp.var.rate      32950 non-null float64
cons.price.idx    32950 non-null float64
cons.conf.idx     32950 non-null float64
euribor3m         32950 non-null float64
nr.employed       32950 non-null float64
y                 32950 non-null object
dtypes: float64(5), int64(5), object(11)
memory usage: 5.3+ MB
None
```

Specify parameters for data preprocessing

In [ ]:
```
# the data will be mapped to interval [min_Data,max_Data]. Usually this will b
e [0,1] but you can also specify other values.
max_Data = 0.7
min_Data = 0.3
# list of categorical variables
list_cat = np.array([0,1,2,3,4,5,6,7])
# list of numerical variables
list_flt = np.array([8,9,10,11,12,13,14,15,16,17,18,19,20])
# among numerical variables, which ones are discrete. This is referred as cont
inuous-discrete variables in Appendix C.1.3 in our paper.
# Examples include variables that take integer values, for example month, day
 of week, number of custumors etc. Other examples include numerical variables
 that are recorded on a discrete grid (for example salary).
list_discrete = np.array([8,9])
```

Data pre-processing (decompressing)

Here we basically do two things. First, the raw data matrix is sorted and normalized (squashed). Then, we will "decompress" the categorical variables in the raw data matrix into one-hot encoding features. In our implementations, we always use the suffix `_decomressed` to indicate any variables that are based on one-hot representations for categorical variables. For example, `Mask_decompressed` is the missingness indicator of the data matrix after decompressed into one-hot encodings.

In [4]:
```
# sort the variables in the data matrix, so that categorical variables appears
first. The resulting data matrix is Data_sub
list_discrete_in_flt = (np.in1d(list_flt, list_discrete).nonzero()[0])
list_discrete_compressed = list_discrete_in_flt + len(list_cat)

if len(list_flt)>0 and len(list_cat)>0:
    list_var = np.concatenate((list_cat,list_flt))
elif len(list_flt)>0:
    list_var = list_flt
```

```
else:
    list_var = list_cat
Data_sub = Data[:,list_var]
dic_var_type = np.zeros(Data_sub.shape[1])
dic_var_type[0:len(list_cat)] = 1

# In this notebook we assume the raw data matrix is fully observed
Mask = np.ones(Data_sub.shape)
# Normalize/squash the data matrix
Data_std = (Data_sub - Data_sub.min(axis=0)) / (Data_sub.max(axis=0) - Data_su
b.min(axis=0))
scaling_factor = (Data_sub.max(axis=0) - Data_sub.min(axis=0))/(max_Data - min
_Data)
Data_sub = Data_std * (max_Data - min_Data) + min_Data

# decompress categorical data into one hot representation
Data_cat = Data[:,list_cat].copy()
Data_flt = Data[:,list_flt].copy()
Data_compressed = np.concatenate((Data_cat,Data_flt),axis = 1)
Data_decompressed, Mask_decompressed, cat_dims, DIM_FLT = process.data_preproc
ess(Data_sub,Mask,dic_var_type)
Data_train_decompressed, Data_test_decompressed, mask_train_decompressed, mask
_test_decompressed,mask_train_compressed, mask_test_compressed,Data_train_comp
ressed, Data_test_compressed = train_test_split(
        Data_decompressed, Mask_decompressed,Mask,Data_compressed,test_size=0.
1, random_state=rs)

list_discrete = list_discrete_in_flt + (cat_dims.sum()).astype(int)

Data_decompressed = np.concatenate((Data_train_decompressed, Data_test_decompr
essed), axis=0)
Data_train_orig = Data_train_decompressed.copy()
Data_test_orig = Data_test_decompressed.copy()

# Note that here we have added some noise to continuous-discrete variables to
 help training. Alternatively, you can also disable this by changing the noise
ratio to 0.
Data_noisy_decompressed,records_d, intervals_d = process.noisy_transform(Data_
decompressed, list_discrete, noise_ratio = 0.99)
noise_record = Data_noisy_decompressed - Data_decompressed
Data_train_noisy_decompressed = Data_noisy_decompressed[0:Data_train_decompres
sed.shape[0],:]
Data_test_noisy_decompressed = Data_noisy_decompressed[Data_train_decompressed
.shape[0]:,:]
```

```
(32950, 21)
```

Load or Train a VAEM model

In [5]:

```
import utils.active_learning as active_learning
vae = active_learning.p_vae_active_learning(Data_train_compressed, Data_train_
noisy_decompressed,mask_train_decompressed,Data_test_decompressed,mask_test_co
mpressed,mask_test_decompressed,cat_dims,DIM_FLT,dic_var_type,args)
```

```
Tensor("is/generator/mul_5:0", shape=(?, 124), dtype=float32)
Tensor("is/generator/mul_43:0", shape=(?, 124), dtype=float32)

/home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/ops/g
radients_impl.py:96: UserWarning: Converting sparse IndexedSlices to a dense T
ensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "

INFO:tensorflow:Restoring parameters from ./saved_weights/bank3_gen/encoder.te
nsorflow
INFO:tensorflow:Restoring parameters from ./saved_weights/bank3_gen/generator.
tensorflow
```

```
WARNING:tensorflow:From /home/paperspace/Desktop/VAEM NIPS/models/model.py:55
4: all_variables (from tensorflow.python.ops.variables) is deprecated and will
be removed after 2017-03-02.
Instructions for updating:
Please use tf.global_variables instead.
```

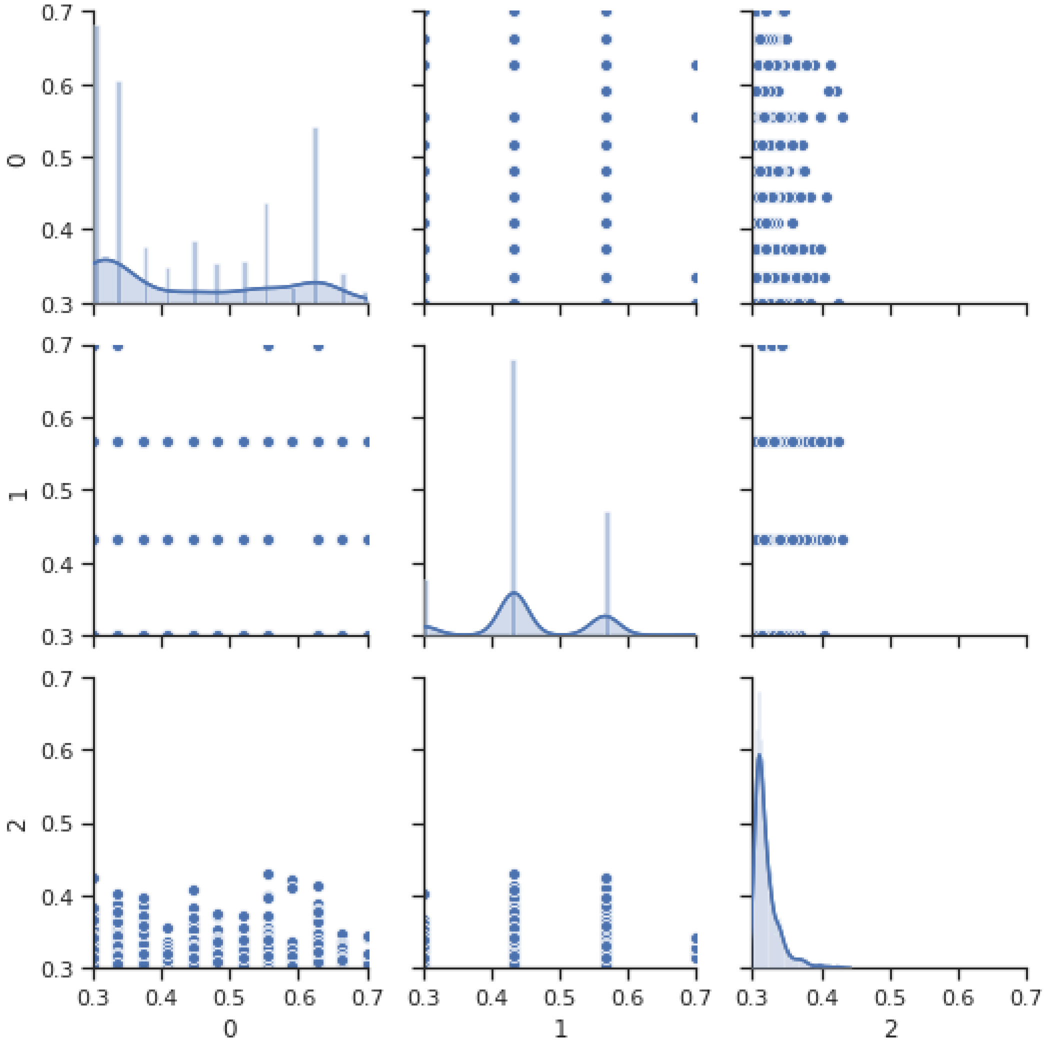
Use the model to impute data and generate pairplots

In [6]:

```
tf.reset_default_graph()
### Impute missing data. Fthe mask to be zeros

x_recon,z_posterior,x_recon_cat_p = vae.get_imputation( Data_train_noisy_decom
pressed, mask_train_decompressed*0,cat_dims,dic_var_type) ## one hot already c
pverted to integer


x_real = process.compress_data(Data_train_decompressed,cat_dims, dic_var_type)
## x_real still needs conversion
x_real_cat_p = Data_train_decompressed[:,0:(cat_dims.sum()).astype(int)]


# max_Data = 0.7
# min_Data = 0.3
Data_std = (x_real - x_real.min(axis=0)) / (x_real.max(axis=0) - x_real.min(ax
is=0))
scaling_factor = (x_real.max(axis=0) - x_real.min(axis=0))/(max_Data - min_Dat
a)
Data_real = Data_std * (max_Data - min_Data) + min_Data

fast_plot = 1

sub_id = [1,2,10]

if fast_plot ==1:
    Data_real = pd.DataFrame(Data_real[:,sub_id])
    g = sns.pairplot(Data_real.sample(min(1000,x_real.shape[0])),diag_kind =
'kde')
    g = g.map_diag(sns.distplot, bins = 50,norm_hist = True)
    g.set(xlim=(min_Data,max_Data), ylim = (min_Data,max_Data))
else:
    Data_real = pd.DataFrame(Data_real[:,sub_id])
    g = sns.pairplot(Data_real.sample(min(10000,x_real.shape[0])),diag_kind =
'kde')
    g = g.map_diag(sns.distplot, bins = 50,norm_hist = True)
    g = g.map_upper(plt.scatter,marker='+')
    g = g.map_lower(sns.kdeplot, cmap="hot",shade=True,bw=.1)
    g.set(xlim=(min_Data,max_Data), ylim = (min_Data,max_Data))
```

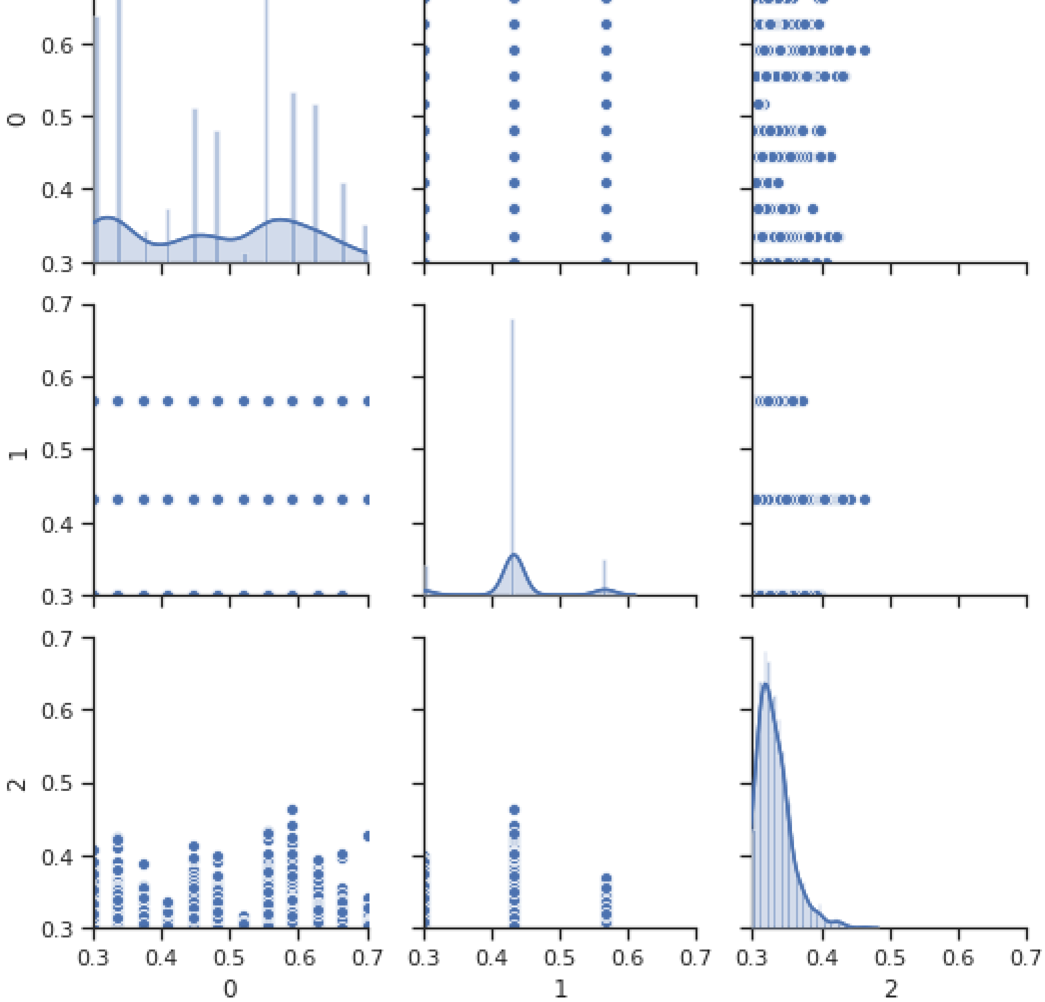
```
76.0

/home/paperspace/Desktop/VAEM NIPS/models/model.py:1087: RuntimeWarning: inval
id value encountered in true_divide
  decoded_cat_int_p = decoded_cat_int_p/np.sum(decoded_cat_int_p,1,keepdims=Tr
ue)
/home/paperspace/Desktop/VAEM NIPS/models/model.py:1099: RuntimeWarning: inval
id value encountered in less
  decoded_cat_int[n,d] = np.random.choice(len(decoded_cat_int_p[n,:]), 1 , p=d
ecoded_cat_int_p[n,:])

11.0
2.0
7.0
1.0
2.0
0.0
1.0
```

```
76.0
11.0
3.0
7.0
2.0
2.0
2.0
1.0
```



In [7]:

```
Data_fake_noisy= x_recon
Data_fake = process.invert_noise(Data_fake_noisy,list_discrete_compressed,reco
rds_d)

Data_std = (Data_fake - x_real.min(axis=0)) / (x_real.max(axis=0) - x_real.min
(axis=0))
Data_fake = Data_std * (max_Data - min_Data) + min_Data


sub_id = [1,2,10]

if fast_plot ==1:
    g = sns.pairplot(pd.DataFrame(Data_fake[:,sub_id]).sample(min(1000,x_real.
shape[0])),diag_kind = 'kde')
    g = g.map_diag(sns.distplot, bins = 50,norm_hist = True)
    g.set(xlim=(min_Data,max_Data), ylim = (min_Data,max_Data))
else:
    g = sns.pairplot(pd.DataFrame(Data_fake[:,sub_id]).sample(min(1000,x_real.
shape[0])),diag_kind = 'kde')
    g = g.map_diag(sns.distplot, bins = 50,norm_hist = True)
    g = g.map_upper(plt.scatter,marker='+')
    g = g.map_lower(sns.kdeplot, cmap="hot",shade=True,bw=.1)
    g.set(xlim=(min_Data,max_Data), ylim = (min_Data,max_Data))
```



Train a discriminator on top of the model and perform SAIA

In [8]:
```
args = params.Params('./hyperparameters/bank_SAIA.json')
import utils.active_learning as active_learning
vae = active_learning.p_vae_active_learning(Data_train_compressed, Data_train_
noisy_decompressed,mask_train_decompressed,Data_test_decompressed,mask_test_co
mpressed,mask_test_decompressed,cat_dims,DIM_FLT,dic_var_type,args)
```

```
Tensor("is/generator/mul_5:0", shape=(?, 124), dtype=float32)
Tensor("is/generator/mul_43:0", shape=(?, 124), dtype=float32)
```

```
/home/paperspace/anaconda3/lib/python3.6/site-packages/tensorflow/python/ops/g
radients_impl.py:96: UserWarning: Converting sparse IndexedSlices to a dense T
ensor of unknown shape. This may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
```

```
INFO:tensorflow:Restoring parameters from ./saved_weights/bank3_gen/encoder.te
nsorflow
INFO:tensorflow:Restoring parameters from ./saved_weights/bank3_gen/generator.
tensorflow
Epoch: 0        negative training ELBO per observed feature: 5081.59, Cat_ter
m: 0.09, Flt_term: -0.57,z_term: 62.83
Epoch: 1        negative training ELBO per observed feature: 2226.58, Cat_ter
m: 0.09, Flt_term: -0.56,z_term: 26.07
Epoch: 2        negative training ELBO per observed feature: 1549.46, Cat_ter
m: 0.09, Flt_term: -0.56,z_term: 16.15
Epoch: 3        negative training ELBO per observed feature: 1267.07, Cat_ter
m: 0.09, Flt_term: -0.59,z_term: 10.53
Epoch: 4        negative training ELBO per observed feature: 1094.83, Cat_ter
m: 0.09, Flt_term: -0.58,z_term: 9.83
Epoch: 5        negative training ELBO per observed feature: 972.37, Cat_term:
0.08, Flt_term: -0.59,z_term: 7.71
Epoch: 6        negative training ELBO per observed feature: 891.73, Cat_term:
0.09, Flt_term: -0.59,z_term: 7.22
Epoch: 7        negative training ELBO per observed feature: 811.60, Cat_term:
```

```
0.09, Flt_term: -0.58,z_term: 6.13
Epoch: 8        negative training ELBO per observed feature: 771.66, Cat_term:
0.09, Flt_term: -0.59,z_term: 5.25
Epoch: 9        negative training ELBO per observed feature: 718.78, Cat_term:
0.09, Flt_term: -0.61,z_term: 4.92
Epoch: 10       negative training ELBO per observed feature: 676.23, Cat_term:
0.09, Flt_term: -0.60,z_term: 3.82
Epoch: 11       negative training ELBO per observed feature: 627.54, Cat_term:
0.08, Flt_term: -0.59,z_term: 3.43
Epoch: 12       negative training ELBO per observed feature: 590.13, Cat_term:
0.09, Flt_term: -0.60,z_term: 2.60
Epoch: 13       negative training ELBO per observed feature: 554.96, Cat_term:
0.08, Flt_term: -0.59,z_term: 2.53
Epoch: 14       negative training ELBO per observed feature: 529.37, Cat_term:
0.08, Flt_term: -0.60,z_term: 1.83
Epoch: 15       negative training ELBO per observed feature: 496.02, Cat_term:
0.09, Flt_term: -0.58,z_term: 1.85
Epoch: 16       negative training ELBO per observed feature: 467.30, Cat_term:
0.08, Flt_term: -0.60,z_term: 1.76
Epoch: 17       negative training ELBO per observed feature: 442.08, Cat_term:
0.08, Flt_term: -0.60,z_term: 1.39
Epoch: 18       negative training ELBO per observed feature: 413.42, Cat_term:
0.09, Flt_term: -0.60,z_term: 1.00
Epoch: 19       negative training ELBO per observed feature: 382.13, Cat_term:
0.09, Flt_term: -0.60,z_term: 0.44
Epoch: 20       negative training ELBO per observed feature: 359.26, Cat_term:
0.08, Flt_term: -0.61,z_term: -0.41
Epoch: 21       negative training ELBO per observed feature: 328.52, Cat_term:
0.08, Flt_term: -0.60,z_term: -0.48
Epoch: 22       negative training ELBO per observed feature: 301.39, Cat_term:
0.08, Flt_term: -0.59,z_term: -0.56
Epoch: 23       negative training ELBO per observed feature: 278.36, Cat_term:
0.08, Flt_term: -0.62,z_term: -1.18
Epoch: 24       negative training ELBO per observed feature: 252.63, Cat_term:
0.08, Flt_term: -0.60,z_term: -1.29
Epoch: 25       negative training ELBO per observed feature: 229.46, Cat_term:
0.08, Flt_term: -0.60,z_term: -2.14
Epoch: 26       negative training ELBO per observed feature: 207.38, Cat_term:
0.08, Flt_term: -0.60,z_term: -2.08
Epoch: 27       negative training ELBO per observed feature: 192.34, Cat_term:
0.08, Flt_term: -0.60,z_term: -2.66
Epoch: 28       negative training ELBO per observed feature: 169.82, Cat_term:
0.08, Flt_term: -0.62,z_term: -2.85
Epoch: 29       negative training ELBO per observed feature: 157.50, Cat_term:
0.08, Flt_term: -0.60,z_term: -2.86
Epoch: 30       negative training ELBO per observed feature: 145.15, Cat_term:
0.08, Flt_term: -0.61,z_term: -3.18
Epoch: 31       negative training ELBO per observed feature: 134.72, Cat_term:
0.08, Flt_term: -0.62,z_term: -3.41
Epoch: 32       negative training ELBO per observed feature: 123.38, Cat_term:
0.08, Flt_term: -0.58,z_term: -3.62
Epoch: 33       negative training ELBO per observed feature: 115.80, Cat_term:
0.08, Flt_term: -0.59,z_term: -3.77
Epoch: 34       negative training ELBO per observed feature: 106.00, Cat_term:
0.08, Flt_term: -0.61,z_term: -3.62
Epoch: 35       negative training ELBO per observed feature: 99.92, Cat_term:
0.08, Flt_term: -0.59,z_term: -4.08
Epoch: 36       negative training ELBO per observed feature: 91.88, Cat_term:
0.08, Flt_term: -0.59,z_term: -3.97
Epoch: 37       negative training ELBO per observed feature: 86.50, Cat_term:
0.08, Flt_term: -0.59,z_term: -4.16
Epoch: 38       negative training ELBO per observed feature: 81.59, Cat_term:
0.08, Flt_term: -0.61,z_term: -4.35
Epoch: 39       negative training ELBO per observed feature: 77.57, Cat_term:
0.08, Flt_term: -0.60,z_term: -4.35
Epoch: 40       negative training ELBO per observed feature: 71.84, Cat_term:
```

```
0.08, Flt_term: -0.59,z_term: -4.44
Epoch: 41       negative training ELBO per observed feature: 68.15, Cat_term:
0.08, Flt_term: -0.59,z_term: -4.48
Epoch: 42       negative training ELBO per observed feature: 63.78, Cat_term:
0.08, Flt_term: -0.59,z_term: -4.52
Epoch: 43       negative training ELBO per observed feature: 60.68, Cat_term:
0.08, Flt_term: -0.58,z_term: -4.65
Epoch: 44       negative training ELBO per observed feature: 56.97, Cat_term:
0.08, Flt_term: -0.60,z_term: -4.69
Epoch: 45       negative training ELBO per observed feature: 53.56, Cat_term:
0.08, Flt_term: -0.60,z_term: -4.82
Epoch: 46       negative training ELBO per observed feature: 51.72, Cat_term:
0.08, Flt_term: -0.58,z_term: -4.91
Epoch: 47       negative training ELBO per observed feature: 49.12, Cat_term:
0.08, Flt_term: -0.61,z_term: -4.92
Epoch: 48       negative training ELBO per observed feature: 47.77, Cat_term:
0.08, Flt_term: -0.58,z_term: -4.92
Epoch: 49       negative training ELBO per observed feature: 44.49, Cat_term:
0.08, Flt_term: -0.59,z_term: -5.01
Epoch: 50       negative training ELBO per observed feature: 42.90, Cat_term:
0.08, Flt_term: -0.60,z_term: -4.98
Epoch: 51       negative training ELBO per observed feature: 40.53, Cat_term:
0.08, Flt_term: -0.60,z_term: -4.97
Epoch: 52       negative training ELBO per observed feature: 39.43, Cat_term:
0.08, Flt_term: -0.60,z_term: -5.05
Epoch: 53       negative training ELBO per observed feature: 37.83, Cat_term:
0.08, Flt_term: -0.60,z_term: -5.07
Epoch: 54       negative training ELBO per observed feature: 35.81, Cat_term:
0.08, Flt_term: -0.59,z_term: -5.13
Epoch: 55       negative training ELBO per observed feature: 35.37, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.12
Epoch: 56       negative training ELBO per observed feature: 32.97, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.17
Epoch: 57       negative training ELBO per observed feature: 32.19, Cat_term:
0.08, Flt_term: -0.59,z_term: -5.17
Epoch: 58       negative training ELBO per observed feature: 30.92, Cat_term:
0.08, Flt_term: -0.59,z_term: -5.29
Epoch: 59       negative training ELBO per observed feature: 30.05, Cat_term:
0.08, Flt_term: -0.59,z_term: -5.25
Epoch: 60       negative training ELBO per observed feature: 28.63, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.24
Epoch: 61       negative training ELBO per observed feature: 28.27, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.31
Epoch: 62       negative training ELBO per observed feature: 26.76, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.28
Epoch: 63       negative training ELBO per observed feature: 25.90, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.30
Epoch: 64       negative training ELBO per observed feature: 24.86, Cat_term:
0.08, Flt_term: -0.59,z_term: -5.28
Epoch: 65       negative training ELBO per observed feature: 23.85, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.30
Epoch: 66       negative training ELBO per observed feature: 23.22, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.28
Epoch: 67       negative training ELBO per observed feature: 21.70, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.37
Epoch: 68       negative training ELBO per observed feature: 21.81, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.36
Epoch: 69       negative training ELBO per observed feature: 21.22, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.39
Epoch: 70       negative training ELBO per observed feature: 20.35, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.41
Epoch: 71       negative training ELBO per observed feature: 19.85, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.40
Epoch: 72       negative training ELBO per observed feature: 19.11, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.45
Epoch: 73       negative training ELBO per observed feature: 18.69, Cat_term:
```

```
0.08, Flt_term: -0.57,z_term: -5.40
Epoch: 74       negative training ELBO per observed feature: 17.85, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.41
Epoch: 75       negative training ELBO per observed feature: 17.78, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.48
Epoch: 76       negative training ELBO per observed feature: 16.90, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.44
Epoch: 77       negative training ELBO per observed feature: 16.15, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.46
Epoch: 78       negative training ELBO per observed feature: 16.13, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.48
Epoch: 79       negative training ELBO per observed feature: 15.27, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.48
Epoch: 80       negative training ELBO per observed feature: 15.17, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.47
Epoch: 81       negative training ELBO per observed feature: 14.02, Cat_term:
0.08, Flt_term: -0.59,z_term: -5.47
Epoch: 82       negative training ELBO per observed feature: 14.06, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.53
Epoch: 83       negative training ELBO per observed feature: 13.93, Cat_term:
0.08, Flt_term: -0.55,z_term: -5.47
Epoch: 84       negative training ELBO per observed feature: 13.57, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.52
Epoch: 85       negative training ELBO per observed feature: 13.43, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.51
Epoch: 86       negative training ELBO per observed feature: 12.77, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.52
Epoch: 87       negative training ELBO per observed feature: 12.31, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.54
Epoch: 88       negative training ELBO per observed feature: 12.19, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.57
Epoch: 89       negative training ELBO per observed feature: 12.14, Cat_term:
0.09, Flt_term: -0.57,z_term: -5.56
Epoch: 90       negative training ELBO per observed feature: 11.91, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.54
Epoch: 91       negative training ELBO per observed feature: 11.51, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.60
Epoch: 92       negative training ELBO per observed feature: 11.43, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.55
Epoch: 93       negative training ELBO per observed feature: 10.81, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.55
Epoch: 94       negative training ELBO per observed feature: 10.57, Cat_term:
0.08, Flt_term: -0.58,z_term: -5.55
Epoch: 95       negative training ELBO per observed feature: 10.16, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.57
Epoch: 96       negative training ELBO per observed feature: 10.36, Cat_term:
0.08, Flt_term: -0.55,z_term: -5.58
Epoch: 97       negative training ELBO per observed feature: 10.10, Cat_term:
0.09, Flt_term: -0.56,z_term: -5.56
Epoch: 98       negative training ELBO per observed feature: 9.51, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.55
Epoch: 99       negative training ELBO per observed feature: 9.36, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.59
Epoch: 100      negative training ELBO per observed feature: 8.92, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.58
Epoch: 101      negative training ELBO per observed feature: 9.13, Cat_term:
0.08, Flt_term: -0.55,z_term: -5.56
Epoch: 102      negative training ELBO per observed feature: 8.73, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.56
Epoch: 103      negative training ELBO per observed feature: 8.58, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.59
Epoch: 104      negative training ELBO per observed feature: 8.76, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.62
Epoch: 105      negative training ELBO per observed feature: 8.69, Cat_term:
0.08, Flt_term: -0.55,z_term: -5.60
Epoch: 106      negative training ELBO per observed feature: 8.00, Cat_term:
```

```
0.08, Flt_term: -0.57,z_term: -5.61
Epoch: 107      negative training ELBO per observed feature: 8.36, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.63
Epoch: 108      negative training ELBO per observed feature: 8.21, Cat_term:
0.08, Flt_term: -0.55,z_term: -5.63
Epoch: 109      negative training ELBO per observed feature: 7.67, Cat_term:
0.08, Flt_term: -0.57,z_term: -5.64
Epoch: 110      negative training ELBO per observed feature: 7.50, Cat_term:
0.09, Flt_term: -0.57,z_term: -5.60
Epoch: 111      negative training ELBO per observed feature: 7.12, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.60
Epoch: 112      negative training ELBO per observed feature: 6.98, Cat_term:
0.08, Flt_term: -0.56,z_term: -5.60
```