In [1]:

```
import json
import os
from tqdm import tqdm_notebook
import gensim
from mermaid.embeddings import Embedding
from mermaid import preprocessing as pp
from mermaid.embeddings.word_index import WordIndex
from mermaid import plotting
from mermaid import models
import numpy as np
import mermaid
```

```
/home/guanxing/venv2/lib/python3.6/site-packages/h5py/__init__.py:36:
FutureWarning: Conversion of the second argument of issubdtype from `f
loat` to `np.floating` is deprecated. In future, it will be treated as
`np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
/home/guanxing/venv2/lib/python3.6/importlib/_bootstrap.py:219: Runtim
eWarning: compiletime version 3.5 of module 'tensorflow.python.framewo
rk.fast_tensor_util' does not match runtime version 3.6
  return f(*args, **kwds)
```

In [2]:

```
def traverse_file_dir(file_path):
    a = os.walk(file_path)

    file_list = []
    file_list2 = []
    for item in a:
        file_list.append(item)
    for item in file_list[0][2]:
        file_list2.append(os.path.join(file_path,item))
    return file_list2

file_path = '/opt/share/dataset/enwiki/'
file_list = traverse_file_dir(file_path)
```

In [3]:

```
big_stream = pp.Stream()

for i in tqdm_notebook(range(6000),total=6000):
    with open (file_list[i]) as file:
        for single_line in file.readlines():
            a = json.loads(single_line)
            b={}
            b.setdefault('d',a['q'])
            b.setdefault('q',a['a'])
            b.setdefault('label',a['label'])
            big_stream.append(pp.Entry(**b))
```

100% 6000/6000 [00:39<00:00, 151.13it/s]

In [4]:

```
len(big_stream)
```

Out[4]:

```
3635176
```

In [4]:

```
new_stream = pp.Stream()
index = 0
for item in big_stream:
    if item['label']==1:
        new_stream.append(item)
        new_stream.append(big_stream[index+1])
    index+=1
```

In [5]:

```
len(new_stream)
```

Out[5]:

```
1199998
```

In [6]:

```
import os

os.environ['CUDA_VISIBLE_DEVICES'] = '0,1'

from mermaid import preprocessing as pp
from mermaid import sessions
from mermaid.factory import Factory

import json
```

In [7]:

```
train_stream, dev_stream, test_stream = new_stream.split()
```

In [8]:

```
class LoadedWordVecEmbedding(Embedding):
    def __init__(self, word_vector_file_path,dim):
        super().__init__()
        self._w2v = gensim.models.KeyedVectors.load_word2vec_format(word_vector_fil
        self._word_index = WordIndex({
            token:self._w2v.vocab[token].index + 2 for token in self._w2v.vocab
        })
        self._dim = dim


    def make_layer(self):
        embedding_matrix = np.zeros((self.vocab_size, self._dim))
        for key, index in self._word_index.items():
            embedding_matrix[index] = self._w2v[key]
        return keras.layers.Embedding(
                self.vocab_size,
                self._dim,
                weights=[embedding_matrix],
                trainable=self._trainable
        )

    @property
    def vocab_size(self):
        return len(self._word_index)+2

    @property
    def word_index(self):
        return self._word_index

    def __getitem__(self, item):
        return self._w2v[item]
```

In [9]:

```
glove_word_vector_embedding = LoadedWordVecEmbedding('/home/guanxing/glove_word_vec
glove_embedding = Factory.Embedding(glove_word_vector_embedding)
```

In [10]:

```
import keras
```

In [12]:

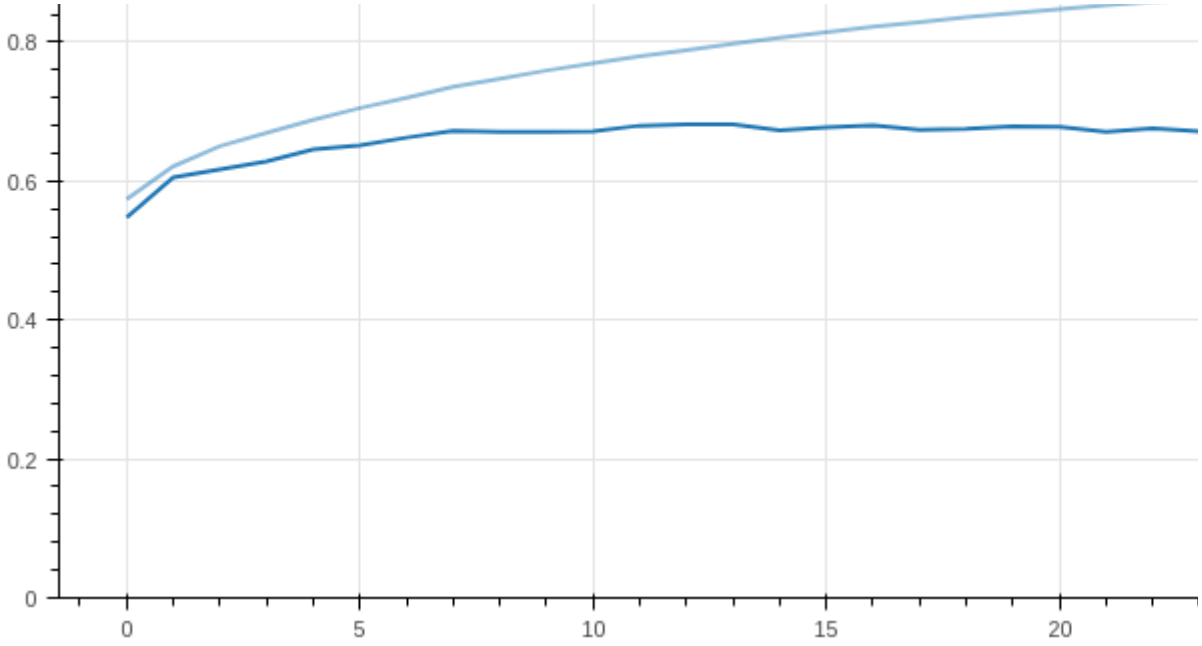
```
sess = sessions.DQTrainingSession()
sess.setup.train(train_stream)
sess.setup.dev(dev_stream)
sess.setup.test(test_stream)
sess.setup.preprocessor('split', 'split')
sess.setup.embedding(glove_embedding, name='glove_embedding')
sess.setup.model(models.ConvKNRM)
sess.setup.fit(epochs=30,batch_size=512)
sess.compile()
sess.model.name='mermaid-convknrm-enwiki'
sess.fit()
sess.eval()

chart = plotting.ModelChart(sess._model.name)

chart.test_on(sess._model, *sess.XY(sess._test))
chart.add_result(sess._fit_history)
plotting.output_notebook()
chart.show()
```

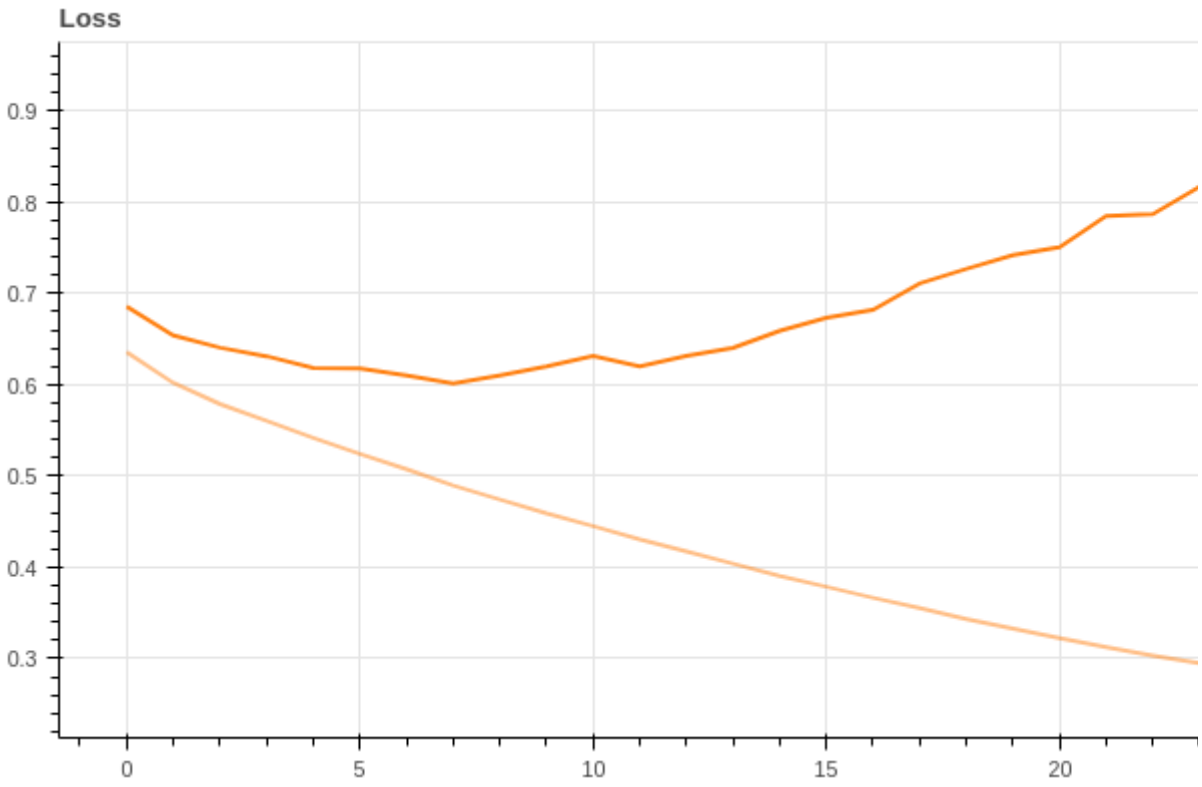
```
Train on 959998 samples, validate on 120000 samples
Epoch 1/30
959998/959998 [==============================] - 120s 125us/step - los
s: 0.6356 - acc: 0.5738 - val_loss: 0.6859 - val_acc: 0.5478
Epoch 2/30
959998/959998 [==============================] - 117s 122us/step - los
s: 0.6019 - acc: 0.6212 - val_loss: 0.6540 - val_acc: 0.6055
Epoch 3/30
959998/959998 [==============================] - 117s 122us/step - los
s: 0.5789 - acc: 0.6503 - val_loss: 0.6407 - val_acc: 0.6164
Epoch 4/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.5604 - acc: 0.6695 - val_loss: 0.6313 - val_acc: 0.6285
Epoch 5/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.5419 - acc: 0.6881 - val_loss: 0.6185 - val_acc: 0.6458
Epoch 6/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.5242 - acc: 0.7050 - val_loss: 0.6177 - val_acc: 0.6511
Epoch 7/30
959998/959998 [==============================] - 117s 122us/step - los
s: 0.5076 - acc: 0.7197 - val_loss: 0.6100 - val_acc: 0.6625
Epoch 8/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.4897 - acc: 0.7356 - val_loss: 0.6013 - val_acc: 0.6718
Epoch 9/30
959998/959998 [==============================] - 118s 122us/step - los
s: 0.4743 - acc: 0.7470 - val_loss: 0.6101 - val_acc: 0.6710
Epoch 10/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.4591 - acc: 0.7590 - val_loss: 0.6200 - val_acc: 0.6711
Epoch 11/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.4451 - acc: 0.7694 - val_loss: 0.6316 - val_acc: 0.6713
Epoch 12/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.4306 - acc: 0.7797 - val_loss: 0.6201 - val_acc: 0.6798
Epoch 13/30
959998/959998 [==============================] - 117s 122us/step - los
```

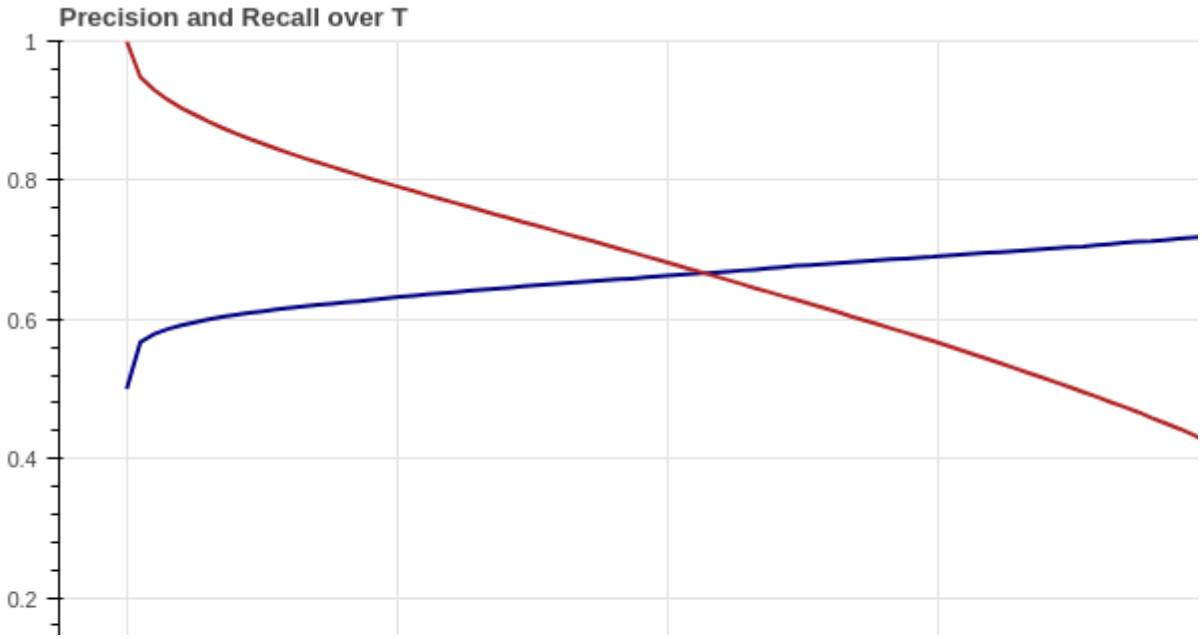
```
s: 0.4175 - acc: 0.7884 - val_loss: 0.6315 - val_acc: 0.6816
Epoch 14/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.4039 - acc: 0.7976 - val_loss: 0.6403 - val_acc: 0.6818
Epoch 15/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.3904 - acc: 0.8064 - val_loss: 0.6590 - val_acc: 0.6730
Epoch 16/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.3786 - acc: 0.8140 - val_loss: 0.6733 - val_acc: 0.6775
Epoch 17/30
959998/959998 [==============================] - 117s 122us/step - los
s: 0.3666 - acc: 0.8218 - val_loss: 0.6820 - val_acc: 0.6804
Epoch 18/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.3551 - acc: 0.8282 - val_loss: 0.7110 - val_acc: 0.6735
Epoch 19/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.3434 - acc: 0.8355 - val_loss: 0.7268 - val_acc: 0.6750
Epoch 20/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.3328 - acc: 0.8416 - val_loss: 0.7418 - val_acc: 0.6787
Epoch 21/30
959998/959998 [==============================] - 117s 122us/step - los
s: 0.3226 - acc: 0.8473 - val_loss: 0.7507 - val_acc: 0.6783
Epoch 22/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.3127 - acc: 0.8531 - val_loss: 0.7848 - val_acc: 0.6708
Epoch 23/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.3033 - acc: 0.8583 - val_loss: 0.7868 - val_acc: 0.6760
Epoch 24/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.2948 - acc: 0.8630 - val_loss: 0.8167 - val_acc: 0.6717
Epoch 25/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.2851 - acc: 0.8681 - val_loss: 0.8249 - val_acc: 0.6746
Epoch 26/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.2774 - acc: 0.8724 - val_loss: 0.8480 - val_acc: 0.6734
Epoch 27/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.2702 - acc: 0.8759 - val_loss: 0.8721 - val_acc: 0.6745
Epoch 28/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.2617 - acc: 0.8805 - val_loss: 0.9171 - val_acc: 0.6653
Epoch 29/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.2547 - acc: 0.8843 - val_loss: 0.9240 - val_acc: 0.6671
Epoch 30/30
959998/959998 [==============================] - 118s 123us/step - los
s: 0.2477 - acc: 0.8876 - val_loss: 0.9408 - val_acc: 0.6672
120000/120000 [==============================] - 10s 84us/step
Loss: 0.9523297865231832, Accuracy: 0.6637666666666666
```

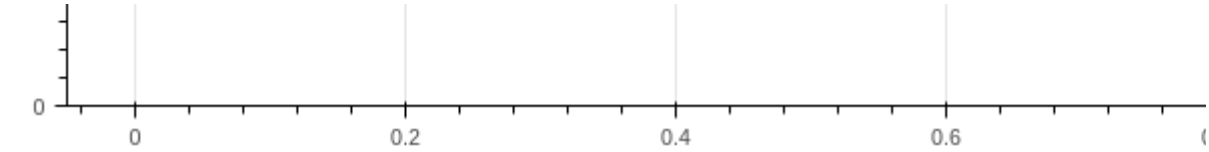
(https://bokeh.pydata.org) Bokeh 0.12.13 successfully loaded.

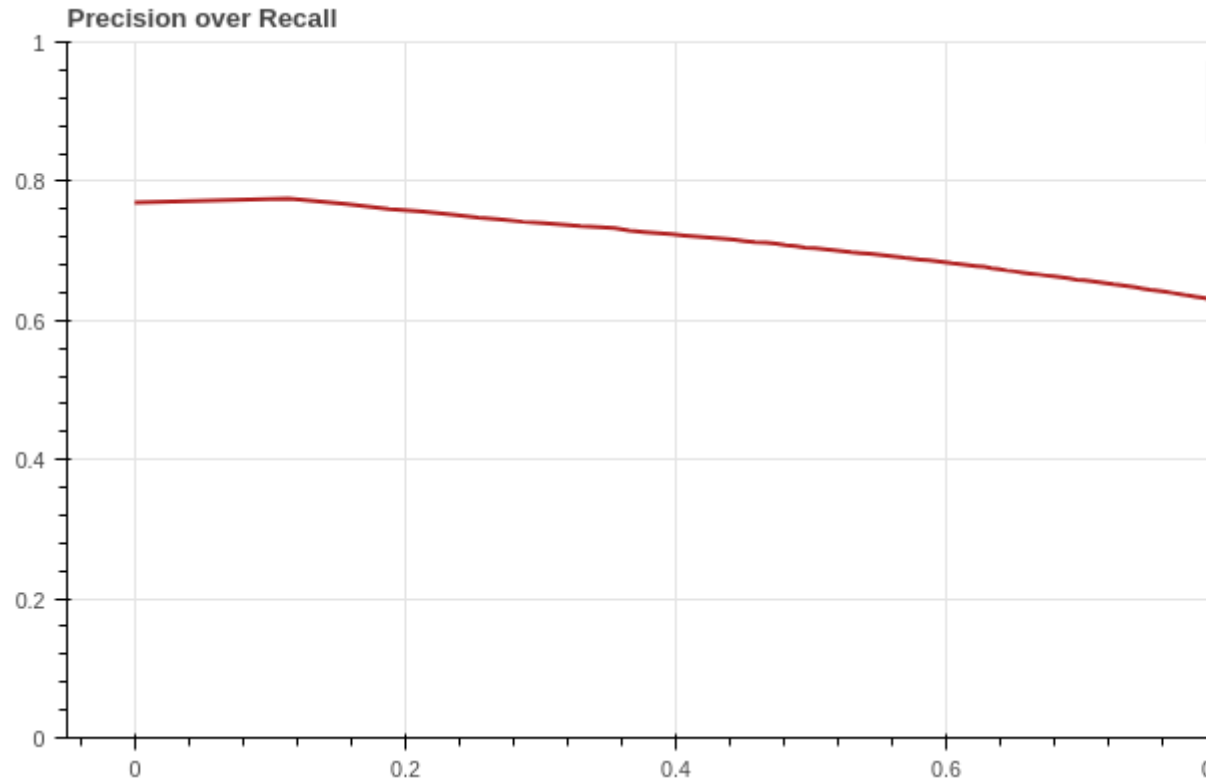


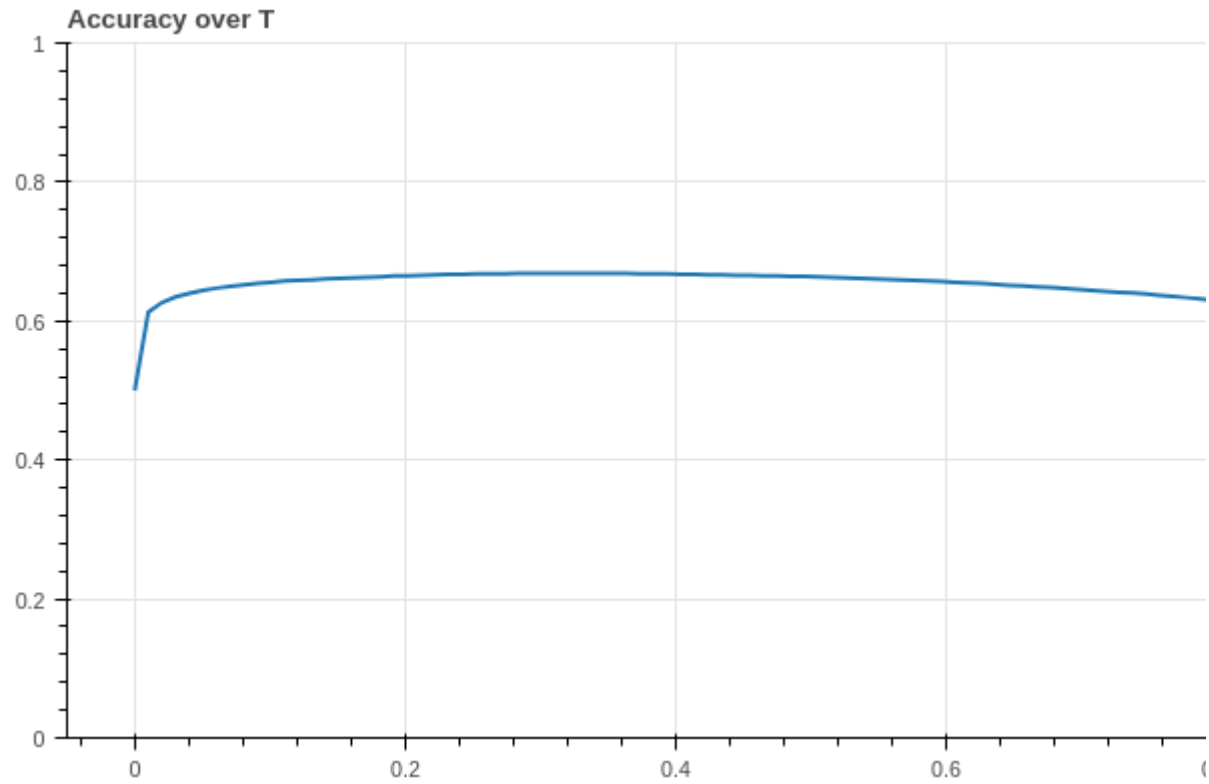

0.8
0.6
0.4
0.2
0
0
5
10
15
20


Loss
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20


Precision and Recall over T
1
0.8
0.6
0.4
0.2


0
0
0.2
0.4
0.6


Precision over Recall
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6


Accuracy over T
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6

In [ ]: