# MXFusion

***Release 1.0***

**Aug 31, 2018**

# Contents

MXFusion is a library for integrating probabilistic modelling with deep learning.

MXFusion helps you rapidly build and test new methods at scale, by focusing on the modularity of probabilistic models and their integration with modern deep learning techniques.

# CHAPTER 1

# Installation

## 1.1 Dependencies / Prerequisites

MXFusion's primary dependencies are MXNet >= 1.2 and Networkx >= 2.1. See requirements.

## 1.2 Supported Architectures / Versions

MXFusion is tested on Python 3.5+ on MacOS and Amazon Linux.

## 1.3 pip

If you just want to use MXFusion and not modify the source, you can install through pip:

```
pip install mxfusion
```

## 1.4 From source

To install MXFusion from source, after cloning the repository run the following from the top-level directory:

```
pip install .
```

# CHAPTER 2

# Design Overview

## 2.1 Topical Guides

Working in MXFusion breaks up into two primary phases. Model definition involves defining the variables, distributions, and functions that make up your model. Inference then takes in real values and learns parameters for your model or gives predictions over the data.

### 2.1.1 Model Definition

MXFusion is a library for doing probabilistic modelling.

Probabilistic Models can be categorized into directed graphical models (DGM, Bayes Net) and undirected graphical models (UGM). Most popular probabilistic models are DGMs, so MXFusion currently only supports DGMs.

A DGM can be fully defined using 3 basic components: deterministic functions, probabilistic distributions, and random variables. As such, those are the primary ModelComponents in MXFusion.

#### Model

The primary data structure of MXFusion is the FactorGraph. FactorGraphs contain Variables and Factors. The FactorGraph exposes methods that Inference algorithms call such as drawing samples from the FactorGraph or computing the log pdf of a set of Variables contained in the FactorGraph.

When you want to start modelling, construct a Model object and start attaching ModelComponents to it. You can then see the Model's components by

```
m = Model()
m.v = Variable()
print(m.components)
```

When a ModelComponent is attached to a Model, it is automatically updated in the Model's internal data structures and will be included in any subsequent inference operations over the model.

### Model Components

All ModelComponents in MXFusion are identified uniquely by a UUID.

###Variables In a model, there are typically four types of variables: a random variable following a probabilistic distribution, a variable which is the outcome of a deterministic function, a parameter (with no prior distribution), and a constant. The definitions of first two types of variables will be discussed later. The latter two types of variables can be defined with the following statement:

```
m.v = Variable(shape=(10,), constraint=PositiveTransformation())
```

At this stage, you do not need to specify whether *v* is a parameter or constant, because, if it is a constant, its value will be provided during inference, otherwise it will be treated as a parameter.

A typical example of when a constant would be specified at inference time is the size (shape) of an observed variable, which is known when data is provided. In the above example, we specify the name of the variable, the shape of the variable and the constraint that the variable has. It defines a 10-dimension vector whose values are always positive (v>=0).

### Factors

####Distributions In a probabilistic model, random variables relate to each other through probabilistic distributions.

During model definition, the typical interface to generate a 2 dimensional random variable *x* from a zero mean unit variance Gaussian distribution looks like:

```
m.x = Normal.generate_variable(mean=0, variance=1, shape=(2,))
```

The two dimensions are independent to each other and both follow the same Gaussian distribution. The parameters or shape of a distribution can also be variables, for example:

```
m.mean = Variable(shape=(2,))
m.y_shape = Variable()
m.y = Normal.generate_variable(mean=m.mean, variance=1, shape=m.y_shape)
```

MXFusion also allows users to specify a prior distribution over pre-existing variables. This is particularly handy for interfacing with neural networks in MXNet because it allows you to set priors over parameters in an existing Gluon Block, such as a neural network implementation. The API for specifying a prior distribution looks like:

```
m.x = Variable(shape=(2,))
m.x.set_prior(Gaussian(mean=0, variance=1))
```

The above code defines a variable *x* and sets the prior distribution of each dimension of *x* to be a scalar unit Gaussian distribution.

Because Models are FactorGraphs, it is common to want to know what ModelComponents come before or after a particular component in the graph. These are accessed through the ModelComponent properties `successors` and `predecessors`.

```
m.mean = Variable()
m.var = Variable()
m.y = Normal.generate_variable(mean=m.mean, variance=m.var)
```

####Functions The last building block of probabilistic models are deterministic functions. The ability to define sophisticated functions allows users to build expressive models with a family of standard probabilistic distributions. As MXNet already provides full functionality for defining a function and automatically evaluating its gradients, Functions in MXFusion are a wrapper over the functions in MXNet's Gluon interface. Functions are defined in standard MXNet syntax and provided to the MXFusion Function wrapper as below.

First we define a function in MXNet Gluon syntax using a Block object:

```
class F(mx.gluon.Block):
    def forward(self, x, y):
        return x*2+y
```

Then we create an MXFusion Function instance by passing in our Gluon function instance:

```
f_gluon = F()
m.f_mf = MXFusionGluonFunction(f_gluon)
```

Then this MXFusion Function can be called using MXFusion variables and its outcome will another variable[s] representing the outcome of the function:

```
m.x = Variable(shape=(2,))
m.y = Variable(shape=(2,))
m.f = f_mf(x, y)
```

#### FAQ

- Why don't you support undirected graphical models (UGM)?
- A UGM is typically defined in terms of a set of potential functions. Each potential function is a non-negative function that is defined on a subset of variables in a model. The joint probability distribution of an UGM is defined as the product of all the potential functions divided by a normalization term (known as a partition function).
- The notation of a DGM and an UGM can be unified into a factor graph, where a factor can be either a probabilistic distribution or a potential function. **In our implementation, the distribution UI is inherited from the factor abstract class, which enables future extension to support UGM** , although inference algorithms for UGM will be completely different.

### 2.1.2 Inference

Notes about inference in MXFusion.

#### Inference Algorithms

MXFusion currently supports stochastic variational inference.

#### Variational Inference

Variational inference is an approximate inference method that can serve as the inference method over generic models built in MXFusion. The main idea of variational inference is to approximate the (often intractable) posterior distribution of our model with a simpler parametric approximation, referred to as a variational posterior distribution. The goal is then to optimize the parameters of this variational posterior distribution to best approximate our true posterior distribution. This is typically done by minimizing the lower bound of the logarithm of the marginal distribution:

\begin{equation} \log p(y|z) = \log \int_x p(y|x) p(x|z) \geq \int_x q(x|y,z) \log \frac{p(y|x) p(x|z)}{q(x|y,z)} = \mathcal{L}(y,z), \label{eqn:lower_bound_1} \end{equation} where $(y|x) p(x|z)$ forms a probabilistic model with $x$ as a latent variable, $q(x|y)$ is the variational posterior distribution, and the lower bound is denoted as $\mathcal{L}(y,z)$. By then taking a natural exponentiation of $\mathcal{L}(y,z)$, we get a lower bound of the marginal probability denoted as $\tilde{p}(y|z) = e^{\mathcal{L}(y,z)}$.

A technical challenge with VI is that the integral of the lower bound of a probabilistic module with respect to external latent variables may not always be tractable. Stochastic variational inference (SVI) offers an approximated solution to this new intractability by applying Monte Carlo Integration. Monte Carlo Integration is applicable to generic probabilistic distributions and lower bounds as long as we are able to draw samples from the variational posterior.

In this case, the lower bound is approximated as \begin{equation} \mathcal{L}(l, z) \approx \frac{1}{N} \sum_i \log \frac{p(l|y_i)e^{\mathcal{L}(y_i,z)}}{q(y_i|z)}, \quad \mathcal{L}(y_i, z) \approx \frac{1}{M} \sum_j \log \frac{p(y_i|x_j) p(x_j|z)}{q(x_j|y_i, z)} , \end{equation} where $y_i|z \sim q(y|z)$, $x_j|y_i,z \sim q(x|y_i,z)$ and $N$ is the number of samples of $y$ and $M$ is the number of samples of $x$ given $y$. Note that if there is a closed form solution of $\tilde{p}(y_i|z)$, the calculation of $\mathcal{L}(y_i,z)$ can be replaced with the closed-form solution.

Let's look at a simple model and then see how we apply stochastic variational inference to it in practice using MXFusion.

### Creating a Posterior

### Examples

### Probabilistic PCA Tutorial

This tutorial will demonstrate Probabilistic PCA, a factor analysis technique.

Maths and notation following Machine Learning: A Probabilistic Perspective.

#### Installation

Follow the instrallation instructions in the README file to get setup.

#### Probabalistic Modeling Introduction

Probabilistic Models can be categorized into directed graphical models (DGM, Bayes Net) and undirected graphical models (UGM). Most popular probabilistic models are DGMs, so MXFusion will only support the definition of DGMs unless there is a strong customer need of UGMs in future.

A DGM can be fully defined using 3 basic components: deterministic functions, probabilistic distributions, and random variables. We show the interface for defining a model using each of the three components below.

First lets import the basic libraries we'll need to train our model and visualize some data.

```
In [1]: import mxfusion as mf
        import mxnet as mx
        import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
/Users/erimeiss/.pyenv/versions/jupyter3/lib/python3.6/site-packages/scipy/stats/morestats.py:12: Dep
  from numpy.testing.decorators import setastest
```

### Data Generation

We'll take as our function to learn components of the log spiral function because it's 2-dimensional and easy to visualize.

```
In [2]: def log_spiral(a,b,t):
            x = a * np.exp(b*t) * np.cos(t)
            y = a * np.exp(b*t) * np.sin(t)
            return np.vstack([x,y]).T
```

We parameterize the function with 100 data points and plot the resulting function.

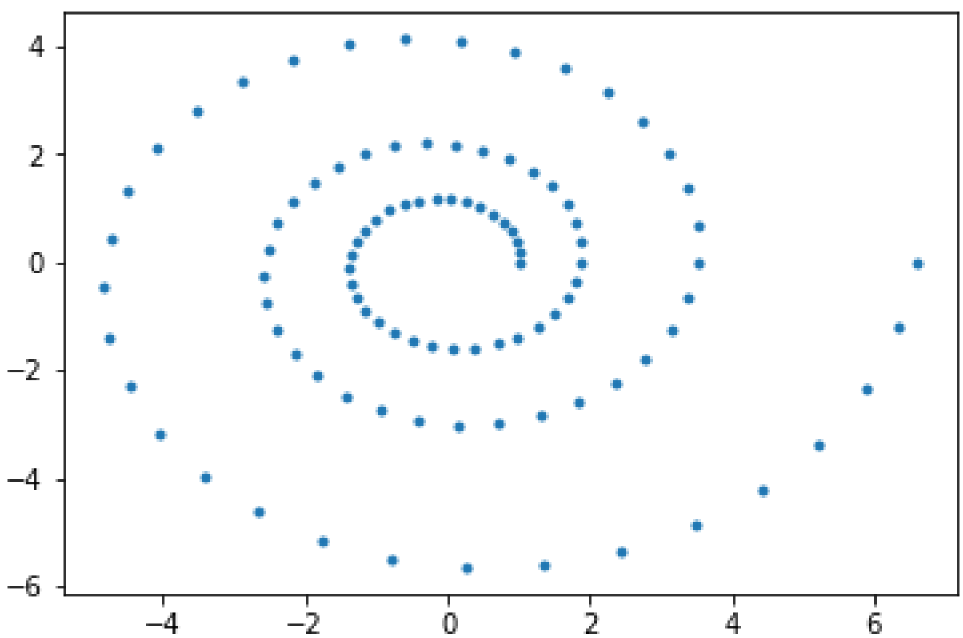
```
In [3]: N = 100
        D = 100
        K = 2

        a = 1
        b = 0.1
        t = np.linspace(0,6*np.pi,N)
        r = log_spiral(a,b,t)
```

```
In [4]: r.shape
```

```
Out[4]: (100, 2)
```

```
In [5]: plt.plot(r[:,0], r[:,1],'.')
```

```
Out[5]: [<matplotlib.lines.Line2D at 0x11ced7668>]
```




We now our project our $K$ dimensional $\mathrm{r}$ into a high-dimensional $D$ space using a random matrix of random weights $W$. Now that $\mathrm{r}$ is embedded in a $D$ dimensional space the goal of PPCA will be to recover $\mathrm{r}$ in it's original low-dimensional $K$ space.

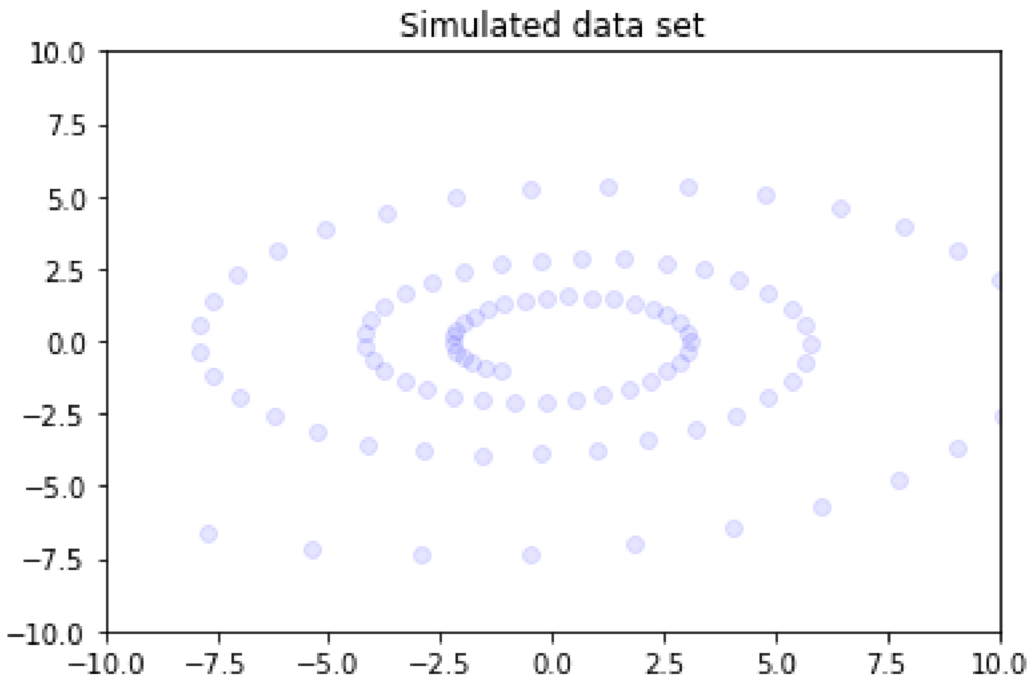
```
In [6]: w = np.random.randn(K,N)
        x_train = np.dot(r,w) + np.random.randn(N,N) * 1e-3
```

```
In [7]: # from sklearn.decomposition import PCA
        # pca = PCA(n_components=2)
```

```
# new_r = pca.fit_transform(r_high)
# plt.plot(new_r[:,0], new_r[:,1],'.')
```

You can explore the higher dimensional data manually by changing `dim1` and `dim2` in the following cell.

```
In [8]: dim1 = 79
        dim2 = 11
        plt.scatter(x_train[:,dim1], x_train[:,dim2], color='blue', alpha=0.1)
        plt.axis([-10, 10, -10, 10])
        plt.title("Simulated data set")
        plt.show()
```




### MXFusion Model Definition

Import MXFusion and MXNet modelling components

```
In [9]: from mxfusion.models import Model
        import mxnet.gluon.nn as nn
        from mxfusion.components import Variable
        from mxfusion.components.variables import PositiveTransformation
```

The primary data structure in MXFusion is the Model. Models hold ModelComponents, such as Variables, Distributions, and Functions which are the what define a probabilistic model.

The model we'll be defining for PPCA is:

$p(z) \sim N(\mu, \mathbf{\Sigma})$

$p(x|z,\theta) \sim N(\mathbf{Wz} + \mu, \Psi)$

where:

$z \in \mathbb{R}^{NxK}, \mu \in \mathbb{R}^K, \mathbf{\Sigma} \in \mathbb{R}^{NxKxK}, x \in \mathbb{R}^{NxD}$

$\Psi \in \mathbb{R}^{NxDxD}, \Psi = [\Psi_0, \dots, \Psi_N], \Psi_i = \sigma^2\mathbf{I}$

$z$ here is our latent variable of interest, $x$ is the observed data, and all other variables are parameters or constants of the model.

First we create an MXFusion Model object to build our PPCA model on.

```
In [10]: m = Model()
```

We attach `Variable` objects to our model to collect them in a centralized place. Internally, these are organized into a factor graph which is used during Inference.

```
In [11]: m.w = Variable(shape=(K,D), initial_value=mx.nd.array(np.random.randn(K,D)))
```

Because the mean of $x$'s distribution is composed of the dot product of $z$ and $W$, we need to create a dot product function. First we create a dot product function in MXNet and then wrap the function into MXFusion using the MXFusionGluonFunction class. `m.dot` can then be called like a normal python function and will apply to the variables it is called on.

```
In [12]: dot = nn.HybridLambda(function='dot')
         m.dot = mf.functions.MXFusionGluonFunction(dot, nOutputs=1, broadcastable=False)
```

Now we define `m.z` which has an identity matrix covariance, `cov`, and zero mean.

`m.z` and `sigma_2` are then used to define `m.x`.

Note that both `sigma_2` and `cov` will be added implicitly into the `Model` because they are inputs to `m.x`.

```
In [13]: cov = mx.nd.broadcast_to(mx.nd.expand_dims(mx.nd.array(np.eye(K,K)), 0),shape=(N,K,K))
         m.z = mf.distributions.MultivariateNormal.define_variable(mean=mx.nd.zeros(shape=(N,K)), cov
         sigma_2 = Variable(shape=(1,), transformation=PositiveTransformation())
         m.x = mf.distributions.Normal.define_variable(mean=m.dot(m.z, m.w), variance=sigma_2, shape=
```

### Posterior Definition

Now that we have our model, we need to define a posterior with parameters for the inference algorithm to optimize. When constructing a Posterior, we pass in the Model it is defined over and ModelComponent's from the original Model are accessible and visible in the Posterior.

The covariance matrix must continue to be positive definite throughout the optimization process in order to succeed in the Cholesky decomposition when drawing samples or computing the log pdf of `q.z`. To satisfy this, we pass the covariance matrix parameters through a Gluon function that forces it into a Symmetric matrix which for suitable initialization values should maintain positive definite-ness throughout the optimization procedure.

```
In [14]: from mxfusion.inference import BatchInferenceLoop, GradBasedInference, StochasticVariational
         class SymmetricMatrix(mx.gluon.HybridBlock):
             def hybrid_forward(self, F, x, *args, **kwargs):
                 return F.sum((F.expand_dims(x, 3)*F.expand_dims(x, 2)), axis=-3)
```

While this model has an analytical solution, we will run Variational Inference to find the posterior to demonstrate inference in a setting where the answer is known.

We place a multivariate normal prior over $z$ because that is $z$'s prior in the model and we don't need to approximate anything in this case. Because the form we're optimizing over is the true model, the optimization is convex and will always converge to the same answer given by classical PCA given enough iterations.

```
In [15]: q = mf.models.Posterior(m)
         sym = mf.components.functions.MXFusionGluonFunction(SymmetricMatrix(), nOutputs=1, broadcast
         cov = Variable(shape=(N,K,K), initial_value=mx.nd.broadcast_to(mx.nd.expand_dims(mx.nd.array
         q.post_cov = sym(cov)
         q.post_mean = Variable(shape=(N,K), initial_value=mx.nd.array(np.random.randn(N,K)))
         q.z.set_prior(mf.distributions.MultivariateNormal(mean=q.post_mean, covariance=q.post_cov))
```

We now take our posterior and model, along with an observation pattern (in our case only `m.x` is observed) and create an inference algorithm. This inference algorithm is combined with a gradient loop to create the Inference method `infr`.

```
In [16]: observed = [m.x]
         alg = StochasticVariationalInference(nSamples=3, model=m, posterior=q, observed=observed)
         infr = GradBasedInference(inference_algorithm=alg,  grad_loop=BatchInferenceLoop())
```

The inference method is then initialized with our training data and we run optimiziation for a while until convergence.

```
In [17]: infr.initialize(x=mx.nd.array(x_train))
```

```
/Users/erimeiss/workspace/MXFusion/src/MXFusion/mxfusion/inference/inference_parameters.py:52: UserWa
  warnings.warn("InferenceParameters has already been initialized.  The existing one will be overwrit
```

```
In [18]: infr.run(max_iter=1000, learning_rate=1e-2, x=mx.nd.array(x_train))
```

```
/Users/erimeiss/workspace/MXFusion/src/MXFusion/mxfusion/inference/inference.py:111: UserWarning: Try
  warnings.warn("Trying to initialize the inference twice, skipping.")
/Users/erimeiss/workspace/MXFusion/src/MXFusion/mxfusion/models/factor_graph.py:194: UserWarning: Fun
  warnings.warn('Function evaluation in FactorGraph.compute_log_prob_RT: the outcome variable '+str(u
```

Once training completes, we retrieve the posterior mean (our trained representation for $\mathbf{Wz} + \mu$) from the inference method and plot it. As shown, the plot recovers (up to rotation) the original 2D data quite well.

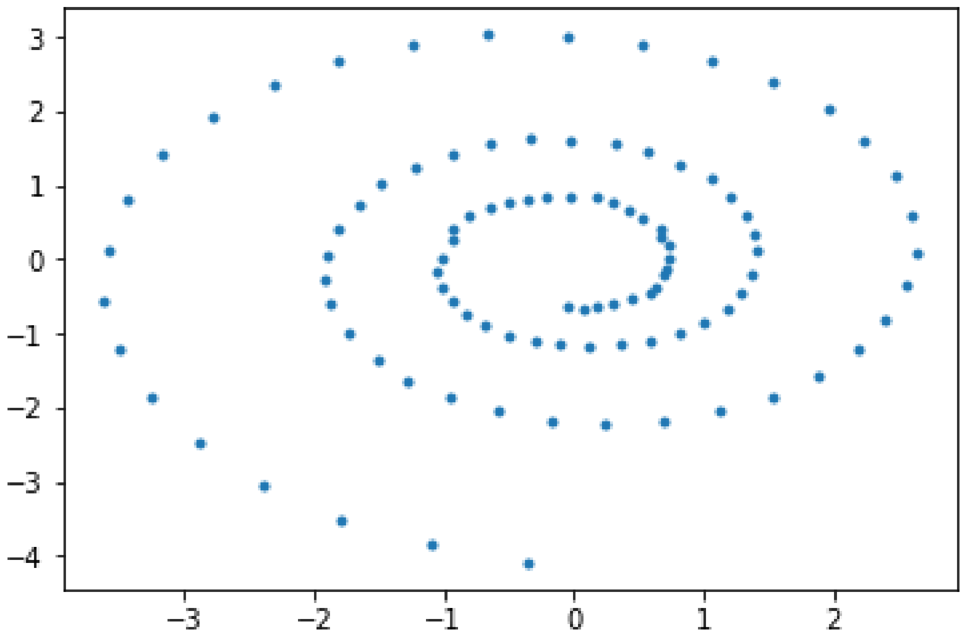
```
In [19]: post_z_mean = infr.params[q.z.factor.mean].asnumpy()
```

```
In [20]: post_z_mean.shape
```

```
Out[20]: (100, 2)
```

```
In [21]: plt.plot(post_z_mean[:,0], post_z_mean[:,1],'.')
```

```
Out[21]: [<matplotlib.lines.Line2D at 0x11d6f3e80>]
```



#### Saving Inference Results

## 2.2 Design Choices

### 2.2.1 CIPs

CIPs are a design proposal mechanism from the Apache Software Foundation.

# CHAPTER 3

# API Reference

| `mxfusion` |
|---|

CHAPTER 4

# Tutorials!

Below is a list of tutorial / example notebooks demonstrating MXFusion's functionality.

## 4.1 Getting Started

## 4.2 Example Models

### 4.2.1 Bayesian Neural Network (VI) for classification (under Development)

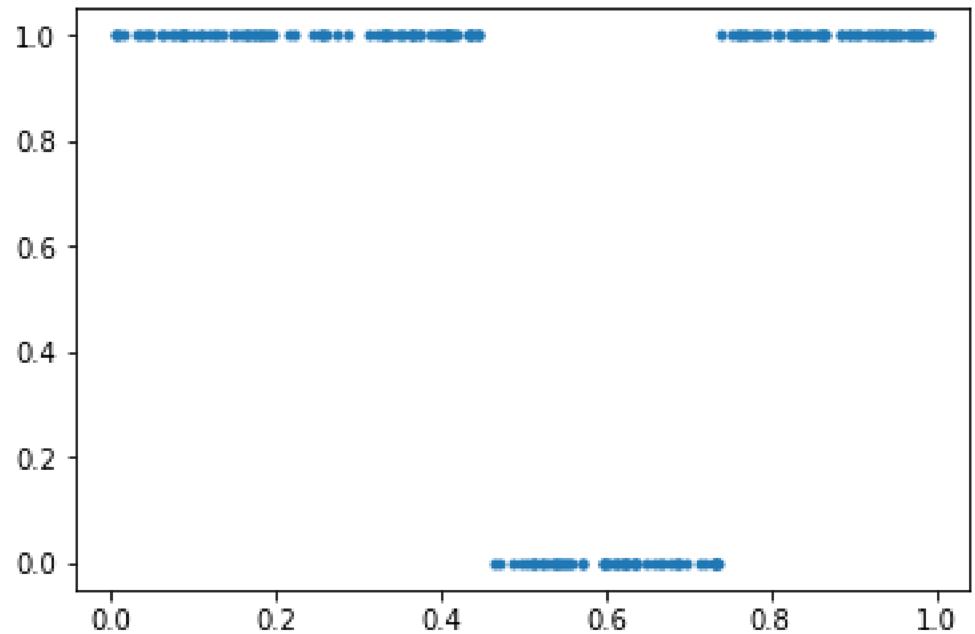
```
In [1]: import mxfusion as mf
        import mxnet as mx
        import numpy as np
        import mxnet.gluon.nn as nn
        import mxfusion.components
        import mxfusion.inference
```

```
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion
  from ._conv import register_converters as _register_converters
```

#### Generate Synthetic Data

```
In [75]: import GPy
         %matplotlib inline
         from pylab import *

         np.random.seed(4)
         k = GPy.kern.RBF(1, lengthscale=0.1)
         x = np.random.rand(200,1)
         y = np.random.multivariate_normal(mean=np.zeros((200,)), cov=k.K(x), size=(1,)).T>0.
         plot(x[:,0], y[:,0], '.')
```

```
Out[75]: [<matplotlib.lines.Line2D at 0x1a22348898>]
```




```
In [76]: D = 10
         net = nn.HybridSequential(prefix='nn_')
         with net.name_scope():
             net.add(nn.Dense(D, activation="tanh", flatten=False, in_units=1))
             net.add(nn.Dense(D, activation="tanh", flatten=False, in_units=D))
             net.add(nn.Dense(2, flatten=False, in_units=D))
         net.initialize(mx.init.Xavier(magnitude=1))
```

```
In [77]: from mxfusion.components.variables.var_trans import PositiveTransformation
         from mxfusion.inference import VariationalPosteriorForwardSampling
```

```
In [78]: m = mf.components.Model()
         m.N = mf.components.Variable()
         m.f = mf.components.functions.MXFusionGluonFunction(net, nOutputs=1, broadcastable=False)
         m.x = mf.components.Variable(shape=(m.N,1))
         m.r = m.f(m.x)
         for _,v in m.r.factor.block_variables:
             v.set_prior(mf.components.distributions.Normal(mean=mx.nd.array([0]),variance=mx.nd.arra
         m.y = mf.components.distributions.Categorical.define_variable(log_prob=m.r,  shape=(m.N,1))
         m.show()
```

```
Variable(45b58) ~ Normal(mean=Variable(738a7), variance=Variable(9ef9b))
Variable(d7aa2) ~ Normal(mean=Variable(b4564), variance=Variable(02ad0))
Variable(ace48) ~ Normal(mean=Variable(3989f), variance=Variable(49e6a))
Variable(35ae3) ~ Normal(mean=Variable(666b4), variance=Variable(5c0d8))
Variable(7a303) ~ Normal(mean=Variable(9d461), variance=Variable(dd20b))
Variable(28ccc) ~ Normal(mean=Variable(37c47), variance=Variable(a7de6))
r = GluonFunctionEvaluation(nn_dense0_weight=Variable(28ccc), nn_dense0_bias=Variable(7a303), nn_dens
y ~ Categorical(log_prob=r)
```

```
In [79]: from mxfusion.inference import BatchInferenceLoop, create_Gaussian_meanfield, GradBasedInfer
```

```
In [80]: observed = [m.y, m.x]
         q = create_Gaussian_meanfield(model=m, observed=observed)
         alg = StochasticVariationalInference(num_samples=5, model=m, posterior=q, observed=observed)
         # alg = MAP(model=m, observed=observed)
         infr = GradBasedInference(inference_algorithm=alg, grad_loop=BatchInferenceLoop())
```

```
In [81]: infr.initialize(y=mx.nd.array(y), x=mx.nd.array(x))
```

```
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/inference/inference_parameters.py:52: UserWarning:In
```

```
In [82]: for v_name, v in m.r.factor.block_variables:
             uuid = v.uuid
             loc_uuid = infr.inference_algorithm.posterior[uuid].factor.variance.uuid
             a = infr.params.param_dict[loc_uuid].data().asnumpy()
             a[:] = 1e-8
             infr.params[infr.inference_algorithm.posterior[uuid].factor.mean] = net.collect_params()
             infr.params[infr.inference_algorithm.posterior[uuid].factor.variance] = mx.nd.array(a)
```

```
In [83]: infr.run(max_iter=500, learning_rate=1e-1, y=mx.nd.array(y), x=mx.nd.array(x), verbose=True)
```

```
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/inference/inference.py:111: UserWarning:Trying to in
```

```
Iteration 1 logL: -1544.5255126953125
Iteration 2 logL: -1539.4837646484375
Iteration 3 logL: -1511.021484375
Iteration 4 logL: -1505.3983154296875
Iteration 5 logL: -1494.5648193359375
Iteration 6 logL: -1491.2451171875
Iteration 7 logL: -1478.662841796875
Iteration 8 logL: -1478.6864013671875
Iteration 9 logL: -1471.1865234375
Iteration 10 logL: -1452.61962890625
Iteration 11 logL: -1451.615478515625
Iteration 12 logL: -1444.682373046875
Iteration 13 logL: -1445.5955810546875
Iteration 14 logL: -1430.4305419921875
Iteration 15 logL: -1418.5712890625
Iteration 16 logL: -1418.8111572265625
Iteration 17 logL: -1404.1358642578125
Iteration 18 logL: -1400.627685546875
Iteration 19 logL: -1381.745361328125
Iteration 20 logL: -1376.2139892578125
Iteration 21 logL: -1372.777099609375
Iteration 22 logL: -1366.9664306640625
Iteration 23 logL: -1361.6920166015625
Iteration 24 logL: -1342.0687255859375
Iteration 25 logL: -1341.3878173828125
Iteration 26 logL: -1335.53271484375
Iteration 27 logL: -1321.6600341796875
Iteration 28 logL: -1322.144287109375
Iteration 29 logL: -1304.44482421875
Iteration 30 logL: -1296.4647216796875
Iteration 31 logL: -1288.6043701171875
Iteration 32 logL: -1288.419189453125
Iteration 33 logL: -1273.767333984375
Iteration 34 logL: -1261.6514892578125
Iteration 35 logL: -1246.7862548828125
Iteration 36 logL: -1237.2008056640625
Iteration 37 logL: -1230.070556640625
Iteration 38 logL: -1217.3485107421875
Iteration 39 logL: -1210.28466796875
Iteration 40 logL: -1196.698486328125
Iteration 41 logL: -1179.746826171875
Iteration 42 logL: -1172.70556640625
Iteration 43 logL: -1153.209716796875
Iteration 44 logL: -1144.113037109375
Iteration 45 logL: -1131.1912841796875
```

```
Iteration 46 logL: -1122.3773193359375
Iteration 47 logL: -1108.426025390625
Iteration 48 logL: -1095.648193359375
Iteration 49 logL: -1082.5948486328125
Iteration 50 logL: -1079.716552734375
Iteration 51 logL: -1066.1383056640625
Iteration 52 logL: -1065.216064453125
Iteration 53 logL: -1048.493896484375
Iteration 54 logL: -1039.2891845703125
Iteration 55 logL: -1040.451416015625
Iteration 56 logL: -1024.9017333984375
Iteration 57 logL: -1007.782958984375
Iteration 58 logL: -1014.7578125
Iteration 59 logL: -991.4736328125
Iteration 60 logL: -991.7649536132812
Iteration 61 logL: -981.59716796875
Iteration 62 logL: -974.068603515625
Iteration 63 logL: -972.3157958984375
Iteration 64 logL: -960.890625
Iteration 65 logL: -958.5018310546875
Iteration 66 logL: -945.4229736328125
Iteration 67 logL: -937.4677124023438
Iteration 68 logL: -921.200439453125
Iteration 69 logL: -915.6636962890625
Iteration 70 logL: -912.5822143554688
Iteration 71 logL: -905.74267578125
Iteration 72 logL: -900.7443237304688
Iteration 73 logL: -883.1107177734375
Iteration 74 logL: -886.2918701171875
Iteration 75 logL: -879.94775390625
Iteration 76 logL: -861.6558837890625
Iteration 77 logL: -854.5107421875
Iteration 78 logL: -851.0504150390625
Iteration 79 logL: -846.66796875
Iteration 80 logL: -838.7744140625
Iteration 81 logL: -826.3006591796875
Iteration 82 logL: -817.10791015625
Iteration 83 logL: -806.6436767578125
Iteration 84 logL: -803.6990966796875
Iteration 85 logL: -794.893310546875
Iteration 86 logL: -792.5907592773438
Iteration 87 logL: -776.2742919921875
Iteration 88 logL: -775.6581420898438
Iteration 89 logL: -767.6826782226562
Iteration 90 logL: -755.8037109375
Iteration 91 logL: -750.8147583007812
Iteration 92 logL: -741.1118774414062
Iteration 93 logL: -739.661376953125
Iteration 94 logL: -724.867431640625
Iteration 95 logL: -721.9619750976562
Iteration 96 logL: -713.5444946289062
Iteration 97 logL: -700.5989379882812
Iteration 98 logL: -692.425537109375
Iteration 99 logL: -685.3265380859375
Iteration 100 logL: -682.6221923828125
Iteration 101 logL: -667.1160888671875
Iteration 102 logL: -671.22021484375
Iteration 103 logL: -657.33984375
Iteration 104 logL: -647.55712890625
```

```
Iteration 105 logL: -638.2933959960938
Iteration 106 logL: -632.623046875
Iteration 107 logL: -632.192626953125
Iteration 108 logL: -622.1544189453125
Iteration 109 logL: -612.350341796875
Iteration 110 logL: -607.607666015625
Iteration 111 logL: -607.3804931640625
Iteration 112 logL: -591.3848876953125
Iteration 113 logL: -579.00537109375
Iteration 114 logL: -574.9884033203125
Iteration 115 logL: -573.00390625
Iteration 116 logL: -558.756103515625
Iteration 117 logL: -556.7438354492188
Iteration 118 logL: -548.1097412109375
Iteration 119 logL: -527.03515625
Iteration 120 logL: -540.4559326171875
Iteration 121 logL: -524.711181640625
Iteration 122 logL: -530.5860595703125
Iteration 123 logL: -521.8486328125
Iteration 124 logL: -502.2843017578125
Iteration 125 logL: -501.84820556640625
Iteration 126 logL: -491.798828125
Iteration 127 logL: -509.27178955078125
Iteration 128 logL: -473.0491943359375
Iteration 129 logL: -490.3575439453125
Iteration 130 logL: -465.9736328125
Iteration 131 logL: -459.7388916015625
Iteration 132 logL: -450.2601318359375
Iteration 133 logL: -440.4290771484375
Iteration 134 logL: -436.97113037109375
Iteration 135 logL: -439.14788818359375
Iteration 136 logL: -414.6134948730469
Iteration 137 logL: -409.9062805175781
Iteration 138 logL: -396.1278076171875
Iteration 139 logL: -414.68865966796875
Iteration 140 logL: -395.8081359863281
Iteration 141 logL: -386.2045593261719
Iteration 142 logL: -378.982666015625
Iteration 143 logL: -368.39642333984375
Iteration 144 logL: -365.70941162109375
Iteration 145 logL: -368.5234680175781
Iteration 146 logL: -360.26531982421875
Iteration 147 logL: -352.19720458984375
Iteration 148 logL: -347.9356689453125
Iteration 149 logL: -353.3387756347656
Iteration 150 logL: -327.9718933105469
Iteration 151 logL: -345.4718017578125
Iteration 152 logL: -333.1092834472656
Iteration 153 logL: -302.427490234375
Iteration 154 logL: -317.42205810546875
Iteration 155 logL: -289.9781799316406
Iteration 156 logL: -296.4144592285156
Iteration 157 logL: -304.27593994140625
Iteration 158 logL: -306.234375
Iteration 159 logL: -284.4197998046875
Iteration 160 logL: -284.7406921386719
Iteration 161 logL: -274.4178466796875
Iteration 162 logL: -267.80206298828125
Iteration 163 logL: -244.25775146484375
```

```
Iteration 164 logL: -254.48374938964844
Iteration 165 logL: -242.91217041015625
Iteration 166 logL: -248.5866241455078
Iteration 167 logL: -250.1661376953125
Iteration 168 logL: -273.66912841796875
Iteration 169 logL: -240.9576416015625
Iteration 170 logL: -256.94140625
Iteration 171 logL: -275.43536376953125
Iteration 172 logL: -214.0106964111328
Iteration 173 logL: -214.67474365234375
Iteration 174 logL: -232.027587890625
Iteration 175 logL: -228.86508178710938
Iteration 176 logL: -240.1166534423828
Iteration 177 logL: -198.66574096679688
Iteration 178 logL: -216.43634033203125
Iteration 179 logL: -207.2248077392578
Iteration 180 logL: -203.01507568359375
Iteration 181 logL: -201.1699676513672
Iteration 182 logL: -247.60006713867188
Iteration 183 logL: -226.33963012695312
Iteration 184 logL: -253.77496337890625
Iteration 185 logL: -191.73776245117188
Iteration 186 logL: -247.42324829101562
Iteration 187 logL: -201.26058959960938
Iteration 188 logL: -177.60061645507812
Iteration 189 logL: -190.0717010498047
Iteration 190 logL: -317.8877868652344
Iteration 191 logL: -215.68914794921875
Iteration 192 logL: -184.3809356689453
Iteration 193 logL: -180.11679077148438
Iteration 194 logL: -205.02462768554688
Iteration 195 logL: -224.42221069335938
Iteration 196 logL: -244.3284912109375
Iteration 197 logL: -224.40420532226562
Iteration 198 logL: -205.4558868408203
Iteration 199 logL: -188.27389526367188
Iteration 200 logL: -218.32017517089844
Iteration 201 logL: -174.22677612304688
Iteration 202 logL: -199.7232208251953
Iteration 203 logL: -157.99331665039062
Iteration 204 logL: -192.910400390625
Iteration 205 logL: -200.99237060546875
Iteration 206 logL: -163.43582153320312
Iteration 207 logL: -213.77777099609375
Iteration 208 logL: -166.8415985107422
Iteration 209 logL: -161.51039123535156
Iteration 210 logL: -176.5122833251953
Iteration 211 logL: -187.05154418945312
Iteration 212 logL: -153.88722229003906
Iteration 213 logL: -188.95333862304688
Iteration 214 logL: -172.976318359375
Iteration 215 logL: -238.98878479003906
Iteration 216 logL: -174.09642028808594
Iteration 217 logL: -168.273681640625
Iteration 218 logL: -189.6393585205078
Iteration 219 logL: -216.9544677734375
Iteration 220 logL: -171.2532196044922
Iteration 221 logL: -174.9771270751953
Iteration 222 logL: -206.28012084960938
```

```
Iteration 223 logL: -191.87185668945312
Iteration 224 logL: -153.0070037841797
Iteration 225 logL: -181.36277770996094
Iteration 226 logL: -169.47288513183594
Iteration 227 logL: -156.15509033203125
Iteration 228 logL: -179.83694458007812
Iteration 229 logL: -183.0515594482422
Iteration 230 logL: -163.0454864501953
Iteration 231 logL: -184.2140350341797
Iteration 232 logL: -219.8138885498047
Iteration 233 logL: -162.65086364746094
Iteration 234 logL: -144.58377075195312
Iteration 235 logL: -179.9770965576172
Iteration 236 logL: -183.8132781982422
Iteration 237 logL: -199.29489135742188
Iteration 238 logL: -151.03273010253906
Iteration 239 logL: -160.94741821289062
Iteration 240 logL: -176.21104431152344
Iteration 241 logL: -188.37820434570312
Iteration 242 logL: -196.54141235351562
Iteration 243 logL: -162.23831176757812
Iteration 244 logL: -141.35182189941406
Iteration 245 logL: -152.73011779785156
Iteration 246 logL: -237.92613220214844
Iteration 247 logL: -210.1531219482422
Iteration 248 logL: -185.01890563964844
Iteration 249 logL: -185.64022827148438
Iteration 250 logL: -221.623046875
Iteration 251 logL: -216.6830596923828
Iteration 252 logL: -160.9233856201172
Iteration 253 logL: -172.77687072753906
Iteration 254 logL: -165.9558868408203
Iteration 255 logL: -201.72703552246094
Iteration 256 logL: -164.8269500732422
Iteration 257 logL: -165.69264221191406
Iteration 258 logL: -183.5315704345703
Iteration 259 logL: -269.22235107421875
Iteration 260 logL: -146.8050994873047
Iteration 261 logL: -212.0876922607422
Iteration 262 logL: -191.9491729736328
Iteration 263 logL: -218.36244201660156
Iteration 264 logL: -162.2360382080078
Iteration 265 logL: -175.89859008789062
Iteration 266 logL: -202.32484436035156
Iteration 267 logL: -160.58258056640625
Iteration 268 logL: -159.10276794433594
Iteration 269 logL: -164.344482421875
Iteration 270 logL: -211.7244873046875
Iteration 271 logL: -192.30337524414062
Iteration 272 logL: -158.49166870117188
Iteration 273 logL: -171.47390747070312
Iteration 274 logL: -183.94886779785156
Iteration 275 logL: -171.7067413330078
Iteration 276 logL: -151.35958862304688
Iteration 277 logL: -173.07603454589844
Iteration 278 logL: -154.04678344726562
Iteration 279 logL: -192.27493286132812
Iteration 280 logL: -192.57839965820312
Iteration 281 logL: -181.84156799316406
```

```
Iteration 282 logL: -185.34580993652344
Iteration 283 logL: -153.8499298095703
Iteration 284 logL: -152.19866943359375
Iteration 285 logL: -144.4599151611328
Iteration 286 logL: -165.732421875
Iteration 287 logL: -176.54833984375
Iteration 288 logL: -164.82205200195312
Iteration 289 logL: -161.06195068359375
Iteration 290 logL: -176.03353881835938
Iteration 291 logL: -151.64808654785156
Iteration 292 logL: -194.75949096679688
Iteration 293 logL: -175.03729248046875
Iteration 294 logL: -165.12147521972656
Iteration 295 logL: -158.97984313964844
Iteration 296 logL: -181.7843780517578
Iteration 297 logL: -162.631591796875
Iteration 298 logL: -177.43496704101562
Iteration 299 logL: -188.93605041503906
Iteration 300 logL: -175.7605743408203
Iteration 301 logL: -159.49876403808594
Iteration 302 logL: -156.8190460205078
Iteration 303 logL: -158.90673828125
Iteration 304 logL: -162.634765625
Iteration 305 logL: -197.76136779785156
Iteration 306 logL: -179.81996154785156
Iteration 307 logL: -160.31796264648438
Iteration 308 logL: -182.8149871826172
Iteration 309 logL: -166.40286254882812
Iteration 310 logL: -161.11827087402344
Iteration 311 logL: -158.80226135253906
Iteration 312 logL: -172.4361114501953
Iteration 313 logL: -171.3463592529297
Iteration 314 logL: -152.3768310546875
Iteration 315 logL: -158.77395629882812
Iteration 316 logL: -213.81350708007812
Iteration 317 logL: -145.558349609375
Iteration 318 logL: -181.20938110351562
Iteration 319 logL: -185.07322692871094
Iteration 320 logL: -207.5948028564453
Iteration 321 logL: -167.41371154785156
Iteration 322 logL: -170.78697204589844
Iteration 323 logL: -158.92723083496094
Iteration 324 logL: -168.30673217773438
Iteration 325 logL: -145.81129455566406
Iteration 326 logL: -171.64073181152344
Iteration 327 logL: -182.69529724121094
Iteration 328 logL: -154.89712524414062
Iteration 329 logL: -151.62460327148438
Iteration 330 logL: -142.93319702148438
Iteration 331 logL: -158.87815856933594
Iteration 332 logL: -133.32958984375
Iteration 333 logL: -152.91761779785156
Iteration 334 logL: -176.51593017578125
Iteration 335 logL: -176.9428253173828
Iteration 336 logL: -195.5206756591797
Iteration 337 logL: -146.91213989257812
Iteration 338 logL: -174.56198120117188
Iteration 339 logL: -157.5927734375
Iteration 340 logL: -163.74888610839844
```

```
Iteration 341 logL: -149.73818969726562
Iteration 342 logL: -141.78402709960938
Iteration 343 logL: -150.82537841796875
Iteration 344 logL: -203.70167541503906
Iteration 345 logL: -234.94883728027344
Iteration 346 logL: -138.6446990966797
Iteration 347 logL: -155.61611938476562
Iteration 348 logL: -170.1373291015625
Iteration 349 logL: -137.4384307861328
Iteration 350 logL: -159.34925842285156
Iteration 351 logL: -189.43511962890625
Iteration 352 logL: -146.205078125
Iteration 353 logL: -162.0035400390625
Iteration 354 logL: -152.1077117919922
Iteration 355 logL: -177.6432342529297
Iteration 356 logL: -162.13482666015625
Iteration 357 logL: -164.792236328125
Iteration 358 logL: -175.9607391357422
Iteration 359 logL: -175.19512939453125
Iteration 360 logL: -153.57888793945312
Iteration 361 logL: -149.2701873779297
Iteration 362 logL: -151.72816467285156
Iteration 363 logL: -151.6085662841797
Iteration 364 logL: -164.00177001953125
Iteration 365 logL: -169.85508728027344
Iteration 366 logL: -174.19601440429688
Iteration 367 logL: -150.11888122558594
Iteration 368 logL: -183.55679321289062
Iteration 369 logL: -175.8426055908203
Iteration 370 logL: -149.37100219726562
Iteration 371 logL: -185.88963317871094
Iteration 372 logL: -151.3091583251953
Iteration 373 logL: -161.74493408203125
Iteration 374 logL: -146.36911010742188
Iteration 375 logL: -159.70030212402344
Iteration 376 logL: -146.5471954345703
Iteration 377 logL: -183.76512145996094
Iteration 378 logL: -162.58302307128906
Iteration 379 logL: -154.88087463378906
Iteration 380 logL: -153.092529296875
Iteration 381 logL: -149.21633911132812
Iteration 382 logL: -159.79013061523438
Iteration 383 logL: -207.46981811523438
Iteration 384 logL: -166.4166717529297
Iteration 385 logL: -163.28904724121094
Iteration 386 logL: -139.99237060546875
Iteration 387 logL: -159.22752380371094
Iteration 388 logL: -147.1323699951172
Iteration 389 logL: -131.47760009765625
Iteration 390 logL: -173.95697021484375
Iteration 391 logL: -175.2364501953125
Iteration 392 logL: -164.6923828125
Iteration 393 logL: -170.76512145996094
Iteration 394 logL: -150.3489532470703
Iteration 395 logL: -149.53738403320312
Iteration 396 logL: -165.0814971923828
Iteration 397 logL: -159.6266326904297
Iteration 398 logL: -205.7377166748047
Iteration 399 logL: -201.95794677734375
```

```
Iteration 400 logL: -173.90322875976562
Iteration 401 logL: -219.3822479248047
Iteration 402 logL: -170.65428161621094
Iteration 403 logL: -154.12339782714844
Iteration 404 logL: -145.6951446533203
Iteration 405 logL: -154.26612854003906
Iteration 406 logL: -168.78125
Iteration 407 logL: -151.6161346435547
Iteration 408 logL: -137.94374084472656
Iteration 409 logL: -162.55181884765625
Iteration 410 logL: -195.5110626220703
Iteration 411 logL: -181.4525146484375
Iteration 412 logL: -176.29464721679688
Iteration 413 logL: -161.6767578125
Iteration 414 logL: -147.3198699951172
Iteration 415 logL: -174.78868103027344
Iteration 416 logL: -241.66427612304688
Iteration 417 logL: -194.82431030273438
Iteration 418 logL: -163.0545196533203
Iteration 419 logL: -142.1942138671875
Iteration 420 logL: -161.17933654785156
Iteration 421 logL: -151.19564819335938
Iteration 422 logL: -270.526123046875
Iteration 423 logL: -174.3603973388672
Iteration 424 logL: -145.0341339111328
Iteration 425 logL: -174.63157653808594
Iteration 426 logL: -134.2644805908203
Iteration 427 logL: -198.86883544921875
Iteration 428 logL: -149.4174346923828
Iteration 429 logL: -144.0459442138672
Iteration 430 logL: -161.4966583251953
Iteration 431 logL: -134.34768676757812
Iteration 432 logL: -134.56263732910156
Iteration 433 logL: -180.6781463623047
Iteration 434 logL: -158.80487060546875
Iteration 435 logL: -159.59298706054688
Iteration 436 logL: -139.64794921875
Iteration 437 logL: -133.45777893066406
Iteration 438 logL: -152.6923370361328
Iteration 439 logL: -156.44227600097656
Iteration 440 logL: -157.8925018310547
Iteration 441 logL: -167.68077087402344
Iteration 442 logL: -161.24176025390625
Iteration 443 logL: -159.16549682617188
Iteration 444 logL: -172.21835327148438
Iteration 445 logL: -156.8822479248047
Iteration 446 logL: -133.2530975341797
Iteration 447 logL: -153.4013214111328
Iteration 448 logL: -133.96762084960938
Iteration 449 logL: -138.84234619140625
Iteration 450 logL: -183.0357666015625
Iteration 451 logL: -170.00534057617188
Iteration 452 logL: -157.46279907226562
Iteration 453 logL: -155.4312286376953
Iteration 454 logL: -178.26963806152344
Iteration 455 logL: -136.466552734375
Iteration 456 logL: -142.8925018310547
Iteration 457 logL: -159.1484832763672
Iteration 458 logL: -173.41558837890625
```

```
Iteration 459 logL: -161.4503936767578
Iteration 460 logL: -165.84906005859375
Iteration 461 logL: -173.14651489257812
Iteration 462 logL: -155.19857788085938
Iteration 463 logL: -174.43728637695312
Iteration 464 logL: -161.63243103027344
Iteration 465 logL: -140.7918701171875
Iteration 466 logL: -161.52334594726562
Iteration 467 logL: -162.0040283203125
Iteration 468 logL: -146.77978515625
Iteration 469 logL: -145.1008758544922
Iteration 470 logL: -145.02806091308594
Iteration 471 logL: -126.25932312011719
Iteration 472 logL: -134.5507049560547
Iteration 473 logL: -172.2655487060547
Iteration 474 logL: -178.7369384765625
Iteration 475 logL: -175.0013885498047
Iteration 476 logL: -155.21022033691406
Iteration 477 logL: -140.81431579589844
Iteration 478 logL: -186.590087890625
Iteration 479 logL: -147.77403259277344
Iteration 480 logL: -167.0510711669922
Iteration 481 logL: -152.2871856689453
Iteration 482 logL: -151.39828491210938
Iteration 483 logL: -129.93521118164062
Iteration 484 logL: -143.72450256347656
Iteration 485 logL: -146.91224670410156
Iteration 486 logL: -127.20298767089844
Iteration 487 logL: -144.4951934814453
Iteration 488 logL: -145.83363342285156
Iteration 489 logL: -132.7200164794922
Iteration 490 logL: -143.2606658935547
Iteration 491 logL: -144.3072509765625
Iteration 492 logL: -148.9021759033203
Iteration 493 logL: -143.65380859375
Iteration 494 logL: -141.4383544921875
Iteration 495 logL: -165.1541290283203
Iteration 496 logL: -136.8163604736328
Iteration 497 logL: -158.55410766601562
Iteration 498 logL: -138.8023681640625
Iteration 499 logL: -158.96397399902344
Iteration 500 logL: -138.33013916015625
```

```
In [12]: for uuid, v in infr.inference_algorithm.posterior.variables.items():
             if uuid in infr.params.param_dict:
                 print(v.name, infr.params[v])
```

```
In [84]: xt = np.linspace(0,1,100)[:,None]
```

```
In [85]: infr2 = VariationalPosteriorForwardSampling(10, [m.x], infr, [m.r])
         res = infr2.run(x=mx.nd.array(xt))
```

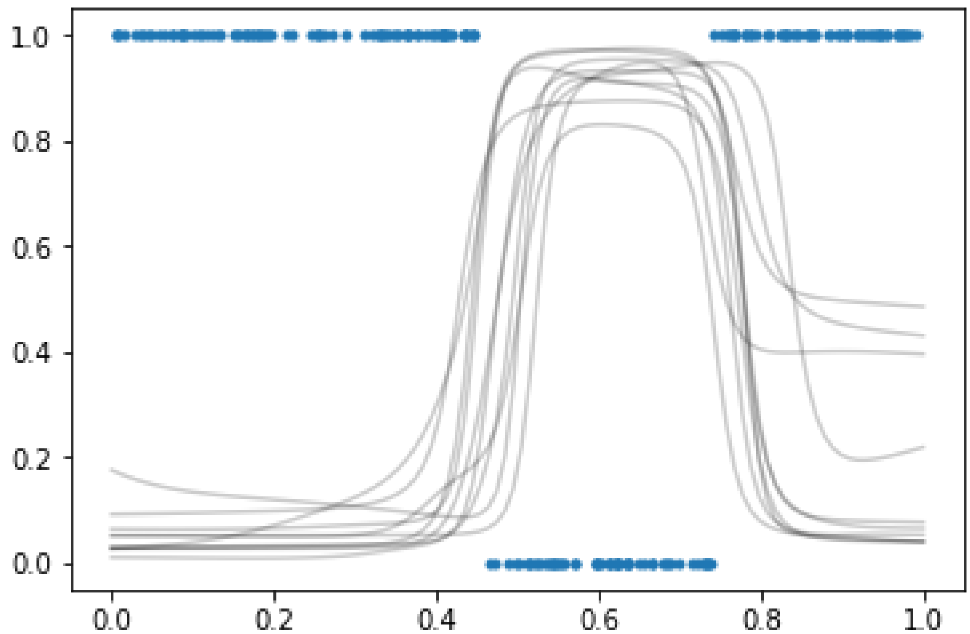
```
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/core/factor_graph.py:65: UserWarning:The value N has
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/core/factor_graph.py:65: UserWarning:The value y has
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/inference/inference_parameters.py:52: UserWarning:In
```

```
In [86]: yt = res[m.r].asnumpy()
```

```
In [87]: # plot(xt[:,0],yt[:,0])
         yt_mean = yt.mean(0)
         yt_std = yt.std(0)
```

```
#plot(xt[:,0], yt.mean(0)[:,0])
#errorbar(xt[:,0],y=yt_mean[:,0],yerr=yt_std[:,0]*2)
for i in range(yt.shape[0]):
    plot(xt[:,0],1./(1+np.exp(-yt[i,:,0])),'k',alpha=0.2)
plot(x[:,0],y[:,0],'.')
```

Out[87]: [<matplotlib.lines.Line2D at 0x1a222e37f0>]




### 4.2.2 Bayesian Neural Network (VI) for regression

#### Zhenwen Dai (2018-8-21)

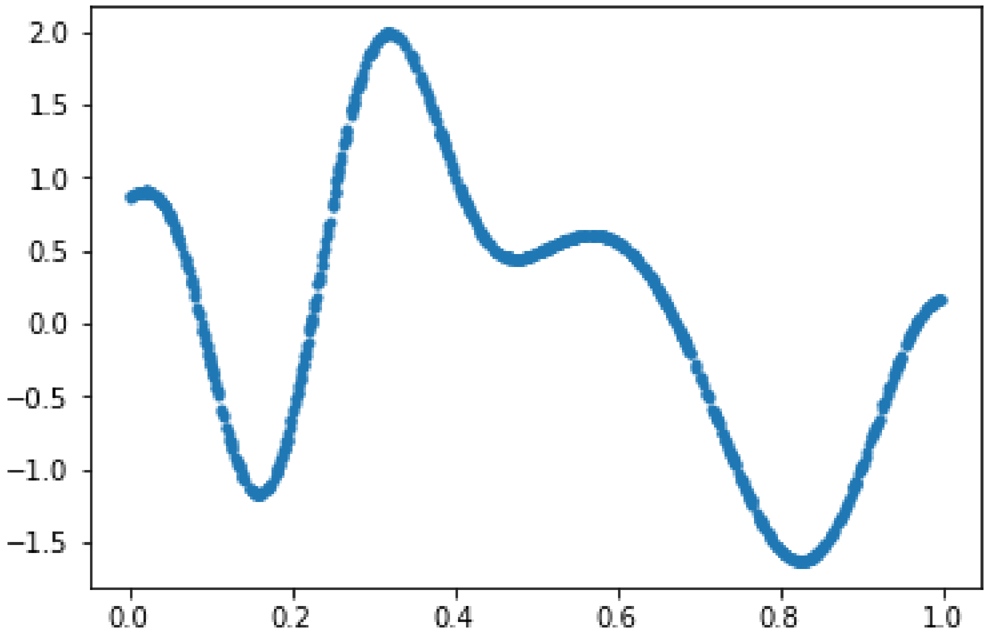
In [1]:
```
import mxfusion as mf
import mxnet as mx
import numpy as np
import mxnet.gluon.nn as nn
import mxfusion.components
import mxfusion.inference
```

```
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion
  from ._conv import register_converters as _register_converters
```

#### Generate Synthetic Data

In [2]:
```
import GPy
%matplotlib inline
from pylab import *

k = GPy.kern.RBF(1, lengthscale=0.1)
x = np.random.rand(1000,1)
y = np.random.multivariate_normal(mean=np.zeros((1000,)), cov=k.K(x), size=(1,)).T
plot(x[:,0], y[:,0], '.')
```

Out[2]: [<matplotlib.lines.Line2D at 0x1097ed358>]




### Model definition

```
In [3]: D = 50
        net = nn.HybridSequential(prefix='nn_')
        with net.name_scope():
            net.add(nn.Dense(D, activation="tanh"))
            net.add(nn.Dense(D, activation="tanh"))
            net.add(nn.Dense(1, flatten=True))
        net.initialize(mx.init.Xavier(magnitude=3))
        _=net(mx.nd.array(x))
```

```
In [4]: from mxfusion.components.var_trans import PositiveTransformation
        from mxfusion.inference import VariationalPosteriorForwardSampling
```

```
In [5]: m = mf.core.Model()
        m.N = mf.components.Variable()
        m.f = mf.components.functions.MXFusionGluonFunction(net, nOutputs=1,broadcastable=False)
        m.x = mf.components.Variable(shape=(m.N,1))
        m.v = mf.components.Variable(shape=(1,), transformation=PositiveTransformation(), initial_val
        #m.prior_variance = mf.core.Variable(shape=(1,), transformation=PositiveTransformation())
        m.r = m.f(m.x)
        for _,v in m.r.factor.block_variables:
            v.set_prior(mf.components.distributions.Normal(mean=mx.nd.array([0]),variance=mx.nd.array
        m.y = mf.components.distributions.Normal.define_variable(mean=m.r, variance=m.v, shape=(m.N,1
        m.show()
```

```
Variable(24825) ~ Normal(mean=Variable(3706e), variance=Variable(78c3f))
Variable(1f4c5) ~ Normal(mean=Variable(d8e05), variance=Variable(b73d6))
Variable(1dbe9) ~ Normal(mean=Variable(6e428), variance=Variable(ee4e8))
Variable(9abb8) ~ Normal(mean=Variable(f8cb1), variance=Variable(73d44))
Variable(cdf2a) ~ Normal(mean=Variable(b4c6f), variance=Variable(5f593))
Variable(e28f4) ~ Normal(mean=Variable(528e0), variance=Variable(c17f7))
r = GluonFunctionEvaluation(nn_dense0_weight=Variable(e28f4), nn_dense0_bias=Variable(cdf2a), nn_dens
y ~ Normal(mean=r, variance=v)
```

### Inference with Meanfield

In [7]:
```
from mxfusion.inference import BatchInferenceLoop, create_Gaussian_meanfield, GradBasedInfere
```

In [9]:
```
observed = [m.y, m.x]
q = create_Gaussian_meanfield(model=m, observed=observed)
alg = StochasticVariationalInference(num_samples=3, model=m, posterior=q, observed=observed)
infr = GradBasedInference(inference_algorithm=alg, grad_loop=BatchInferenceLoop())
```

In [10]:
```
infr.initialize(y=mx.nd.array(y), x=mx.nd.array(x))
```

```
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/inference/inference_parameters.py:52: UserWarning:In
```

In [13]:
```
for v_name, v in m.r.factor.block_variables:
    uuid = v.uuid
    loc_uuid = infr.inference_algorithm.posterior[uuid].factor.variance.uuid
    a = infr.params.param_dict[loc_uuid].data().asnumpy()
    a[:] = 1e-6
    infr.params[infr.inference_algorithm.posterior[uuid].factor.mean] = net.collect_params()
    infr.params[infr.inference_algorithm.posterior[uuid].factor.variance] = mx.nd.array(a)
```

In [14]:
```
infr.run(max_iter=2000, learning_rate=1e-2, y=mx.nd.array(y), x=mx.nd.array(x), verbose=True
```

```
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/inference/inference.py:111: UserWarning:Trying to in
```

```
Iteration 1 logL: -18724.296875
Iteration 2 logL: -18879.15234375
Iteration 3 logL: -18698.25390625
Iteration 4 logL: -18668.005859375
Iteration 5 logL: -18671.970703125
Iteration 6 logL: -18630.625
Iteration 7 logL: -18567.974609375
Iteration 8 logL: -18527.23828125
Iteration 9 logL: -18513.171875
Iteration 10 logL: -18510.267578125
Iteration 11 logL: -18485.630859375
Iteration 12 logL: -18485.716796875
Iteration 13 logL: -18449.76171875
Iteration 14 logL: -18451.197265625
Iteration 15 logL: -18420.908203125
Iteration 16 logL: -18443.97265625
Iteration 17 logL: -18436.333984375
Iteration 18 logL: -18369.080078125
Iteration 19 logL: -18323.1171875
Iteration 20 logL: -18373.0546875
Iteration 21 logL: -18356.80078125
Iteration 22 logL: -18305.6015625
Iteration 23 logL: -18291.322265625
Iteration 24 logL: -18300.39453125
Iteration 25 logL: -18290.212890625
Iteration 26 logL: -18276.2734375
Iteration 27 logL: -18230.921875
Iteration 28 logL: -18253.62109375
Iteration 29 logL: -18226.80078125
Iteration 30 logL: -18193.107421875
Iteration 31 logL: -18209.59765625
Iteration 32 logL: -18201.8828125
Iteration 33 logL: -18127.078125
Iteration 34 logL: -18173.607421875
Iteration 35 logL: -18148.5703125
Iteration 36 logL: -18167.46484375
```

```
Iteration 37 logL: -18109.1640625
Iteration 38 logL: -18074.185546875
Iteration 39 logL: -18067.34765625
Iteration 40 logL: -18076.150390625
Iteration 41 logL: -18047.53515625
Iteration 42 logL: -18040.08984375
Iteration 43 logL: -17991.134765625
Iteration 44 logL: -18030.783203125
Iteration 45 logL: -18033.91796875
Iteration 46 logL: -17993.0703125
Iteration 47 logL: -17993.548828125
Iteration 48 logL: -17983.955078125
Iteration 49 logL: -17946.30859375
Iteration 50 logL: -17986.93359375
Iteration 51 logL: -17938.982421875
Iteration 52 logL: -17913.875
Iteration 53 logL: -17902.4921875
Iteration 54 logL: -17902.658203125
Iteration 55 logL: -17895.572265625
Iteration 56 logL: -17857.169921875
Iteration 57 logL: -17830.033203125
Iteration 58 logL: -17831.1875
Iteration 59 logL: -17793.84375
Iteration 60 logL: -17835.37890625
Iteration 61 logL: -17800.48046875
Iteration 62 logL: -17820.24609375
Iteration 63 logL: -17780.37890625
Iteration 64 logL: -17789.62890625
Iteration 65 logL: -17734.970703125
Iteration 66 logL: -17708.875
Iteration 67 logL: -17703.9453125
Iteration 68 logL: -17700.3359375
Iteration 69 logL: -17677.841796875
Iteration 70 logL: -17666.9609375
Iteration 71 logL: -17661.4296875
Iteration 72 logL: -17669.91015625
Iteration 73 logL: -17585.5234375
Iteration 74 logL: -17579.001953125
Iteration 75 logL: -17597.921875
Iteration 76 logL: -17540.326171875
Iteration 77 logL: -17575.01171875
Iteration 78 logL: -17566.11328125
Iteration 79 logL: -17541.087890625
Iteration 80 logL: -17548.095703125
Iteration 81 logL: -17512.7890625
Iteration 82 logL: -17494.10546875
Iteration 83 logL: -17494.640625
Iteration 84 logL: -17483.634765625
Iteration 85 logL: -17498.158203125
Iteration 86 logL: -17482.9140625
Iteration 87 logL: -17428.267578125
Iteration 88 logL: -17422.775390625
Iteration 89 logL: -17429.705078125
Iteration 90 logL: -17399.8671875
Iteration 91 logL: -17373.126953125
Iteration 92 logL: -17375.6796875
Iteration 93 logL: -17409.5546875
Iteration 94 logL: -17340.95703125
Iteration 95 logL: -17325.294921875
```

```
Iteration 96 logL: -17306.501953125
Iteration 97 logL: -17302.525390625
Iteration 98 logL: -17244.1640625
Iteration 99 logL: -17309.853515625
Iteration 100 logL: -17225.26171875
Iteration 101 logL: -17266.6015625
Iteration 102 logL: -17237.40625
Iteration 103 logL: -17256.572265625
Iteration 104 logL: -17197.884765625
Iteration 105 logL: -17218.955078125
Iteration 106 logL: -17202.572265625
Iteration 107 logL: -17167.3359375
Iteration 108 logL: -17175.16796875
Iteration 109 logL: -17166.728515625
Iteration 110 logL: -17121.45703125
Iteration 111 logL: -17120.2734375
Iteration 112 logL: -17070.46875
Iteration 113 logL: -17085.6953125
Iteration 114 logL: -17067.580078125
Iteration 115 logL: -17064.08203125
Iteration 116 logL: -17076.90625
Iteration 117 logL: -17001.203125
Iteration 118 logL: -17006.65234375
Iteration 119 logL: -17031.591796875
Iteration 120 logL: -16974.060546875
Iteration 121 logL: -16972.890625
Iteration 122 logL: -16949.654296875
Iteration 123 logL: -16929.5078125
Iteration 124 logL: -16937.5078125
Iteration 125 logL: -16895.7734375
Iteration 126 logL: -16890.56640625
Iteration 127 logL: -16884.060546875
Iteration 128 logL: -16840.822265625
Iteration 129 logL: -16881.421875
Iteration 130 logL: -16888.201171875
Iteration 131 logL: -16851.03515625
Iteration 132 logL: -16822.5234375
Iteration 133 logL: -16805.794921875
Iteration 134 logL: -16784.021484375
Iteration 135 logL: -16775.666015625
Iteration 136 logL: -16762.06640625
Iteration 137 logL: -16766.9609375
Iteration 138 logL: -16751.0
Iteration 139 logL: -16740.703125
Iteration 140 logL: -16692.806640625
Iteration 141 logL: -16703.171875
Iteration 142 logL: -16675.4453125
Iteration 143 logL: -16677.59375
Iteration 144 logL: -16631.57421875
Iteration 145 logL: -16644.7109375
Iteration 146 logL: -16621.47265625
Iteration 147 logL: -16644.17578125
Iteration 148 logL: -16604.015625
Iteration 149 logL: -16585.9296875
Iteration 150 logL: -16612.5234375
Iteration 151 logL: -16569.328125
Iteration 152 logL: -16540.49609375
Iteration 153 logL: -16559.5859375
Iteration 154 logL: -16507.060546875
```

```
Iteration 155 logL: -16526.455078125
Iteration 156 logL: -16501.70703125
Iteration 157 logL: -16468.87890625
Iteration 158 logL: -16516.203125
Iteration 159 logL: -16430.58203125
Iteration 160 logL: -16456.365234375
Iteration 161 logL: -16441.638671875
Iteration 162 logL: -16376.8232421875
Iteration 163 logL: -16450.390625
Iteration 164 logL: -16357.951171875
Iteration 165 logL: -16370.08984375
Iteration 166 logL: -16355.0078125
Iteration 167 logL: -16343.19140625
Iteration 168 logL: -16293.0400390625
Iteration 169 logL: -16315.7119140625
Iteration 170 logL: -16316.177734375
Iteration 171 logL: -16303.376953125
Iteration 172 logL: -16262.73828125
Iteration 173 logL: -16268.08984375
Iteration 174 logL: -16257.134765625
Iteration 175 logL: -16230.8056640625
Iteration 176 logL: -16249.55859375
Iteration 177 logL: -16217.859375
Iteration 178 logL: -16235.423828125
Iteration 179 logL: -16164.564453125
Iteration 180 logL: -16165.8466796875
Iteration 181 logL: -16159.5302734375
Iteration 182 logL: -16132.087890625
Iteration 183 logL: -16112.3212890625
Iteration 184 logL: -16146.755859375
Iteration 185 logL: -16114.96484375
Iteration 186 logL: -16080.3515625
Iteration 187 logL: -16062.87109375
Iteration 188 logL: -16100.3974609375
Iteration 189 logL: -16025.275390625
Iteration 190 logL: -16014.99609375
Iteration 191 logL: -16050.162109375
Iteration 192 logL: -16017.828125
Iteration 193 logL: -15968.619140625
Iteration 194 logL: -15986.4677734375
Iteration 195 logL: -15952.431640625
Iteration 196 logL: -15970.521484375
Iteration 197 logL: -15929.3388671875
Iteration 198 logL: -15922.251953125
Iteration 199 logL: -15956.8154296875
Iteration 200 logL: -15898.8701171875
Iteration 201 logL: -15884.9345703125
Iteration 202 logL: -15919.515625
Iteration 203 logL: -15854.3046875
Iteration 204 logL: -15803.6787109375
Iteration 205 logL: -15834.8544921875
Iteration 206 logL: -15821.4169921875
Iteration 207 logL: -15797.1015625
Iteration 208 logL: -15764.2578125
Iteration 209 logL: -15789.982421875
Iteration 210 logL: -15710.556640625
Iteration 211 logL: -15786.0546875
Iteration 212 logL: -15704.74609375
Iteration 213 logL: -15755.013671875
```

```
Iteration 214 logL: -15728.6328125
Iteration 215 logL: -15689.291015625
Iteration 216 logL: -15717.2080078125
Iteration 217 logL: -15675.439453125
Iteration 218 logL: -15646.2705078125
Iteration 219 logL: -15684.91015625
Iteration 220 logL: -15603.919921875
Iteration 221 logL: -15598.0166015625
Iteration 222 logL: -15608.3369140625
Iteration 223 logL: -15613.9365234375
Iteration 224 logL: -15621.728515625
Iteration 225 logL: -15539.240234375
Iteration 226 logL: -15559.958984375
Iteration 227 logL: -15563.345703125
Iteration 228 logL: -15517.5625
Iteration 229 logL: -15474.046875
Iteration 230 logL: -15483.8427734375
Iteration 231 logL: -15477.732421875
Iteration 232 logL: -15464.599609375
Iteration 233 logL: -15416.734375
Iteration 234 logL: -15404.48046875
Iteration 235 logL: -15434.705078125
Iteration 236 logL: -15392.3662109375
Iteration 237 logL: -15370.064453125
Iteration 238 logL: -15381.0703125
Iteration 239 logL: -15363.296875
Iteration 240 logL: -15302.9853515625
Iteration 241 logL: -15333.3662109375
Iteration 242 logL: -15291.2158203125
Iteration 243 logL: -15268.6103515625
Iteration 244 logL: -15222.4453125
Iteration 245 logL: -15184.7666015625
Iteration 246 logL: -15210.71484375
Iteration 247 logL: -15176.96875
Iteration 248 logL: -15098.1201171875
Iteration 249 logL: -15127.9873046875
Iteration 250 logL: -15105.546875
Iteration 251 logL: -15101.3056640625
Iteration 252 logL: -15055.7470703125
Iteration 253 logL: -15006.4853515625
Iteration 254 logL: -15020.3115234375
Iteration 255 logL: -14973.791015625
Iteration 256 logL: -14951.224609375
Iteration 257 logL: -14918.6015625
Iteration 258 logL: -14914.896484375
Iteration 259 logL: -14910.78515625
Iteration 260 logL: -14870.5048828125
Iteration 261 logL: -14838.1015625
Iteration 262 logL: -14849.94140625
Iteration 263 logL: -14783.716796875
Iteration 264 logL: -14784.94921875
Iteration 265 logL: -14826.849609375
Iteration 266 logL: -14770.3994140625
Iteration 267 logL: -14783.13671875
Iteration 268 logL: -14736.634765625
Iteration 269 logL: -14735.66796875
Iteration 270 logL: -14714.548828125
Iteration 271 logL: -14684.767578125
Iteration 272 logL: -14677.47265625
```

```
Iteration 273 logL: -14670.1484375
Iteration 274 logL: -14689.033203125
Iteration 275 logL: -14625.1708984375
Iteration 276 logL: -14618.494140625
Iteration 277 logL: -14611.953125
Iteration 278 logL: -14602.982421875
Iteration 279 logL: -14599.5107421875
Iteration 280 logL: -14534.3681640625
Iteration 281 logL: -14578.837890625
Iteration 282 logL: -14568.205078125
Iteration 283 logL: -14560.39453125
Iteration 284 logL: -14507.1904296875
Iteration 285 logL: -14457.572265625
Iteration 286 logL: -14519.8466796875
Iteration 287 logL: -14471.111328125
Iteration 288 logL: -14444.412109375
Iteration 289 logL: -14447.892578125
Iteration 290 logL: -14415.537109375
Iteration 291 logL: -14445.47265625
Iteration 292 logL: -14430.458984375
Iteration 293 logL: -14357.8017578125
Iteration 294 logL: -14363.615234375
Iteration 295 logL: -14324.3271484375
Iteration 296 logL: -14360.486328125
Iteration 297 logL: -14336.716796875
Iteration 298 logL: -14328.787109375
Iteration 299 logL: -14285.6396484375
Iteration 300 logL: -14297.033203125
Iteration 301 logL: -14277.7119140625
Iteration 302 logL: -14263.6650390625
Iteration 303 logL: -14270.134765625
Iteration 304 logL: -14249.5703125
Iteration 305 logL: -14223.5625
Iteration 306 logL: -14236.5244140625
Iteration 307 logL: -14187.0439453125
Iteration 308 logL: -14192.380859375
Iteration 309 logL: -14183.572265625
Iteration 310 logL: -14122.916015625
Iteration 311 logL: -14112.1806640625
Iteration 312 logL: -14130.890625
Iteration 313 logL: -14094.8515625
Iteration 314 logL: -14065.419921875
Iteration 315 logL: -14069.4033203125
Iteration 316 logL: -14092.4521484375
Iteration 317 logL: -14052.1767578125
Iteration 318 logL: -14037.353515625
Iteration 319 logL: -14007.392578125
Iteration 320 logL: -13996.2666015625
Iteration 321 logL: -14020.9921875
Iteration 322 logL: -14010.9091796875
Iteration 323 logL: -13950.482421875
Iteration 324 logL: -13954.7060546875
Iteration 325 logL: -13945.5439453125
Iteration 326 logL: -13891.216796875
Iteration 327 logL: -13904.3828125
Iteration 328 logL: -13922.8408203125
Iteration 329 logL: -13861.212890625
Iteration 330 logL: -13867.7392578125
Iteration 331 logL: -13853.892578125
```

```
Iteration 332 logL: -13851.076171875
Iteration 333 logL: -13825.45703125
Iteration 334 logL: -13807.490234375
Iteration 335 logL: -13760.224609375
Iteration 336 logL: -13814.10546875
Iteration 337 logL: -13739.6455078125
Iteration 338 logL: -13743.478515625
Iteration 339 logL: -13767.3759765625
Iteration 340 logL: -13725.15625
Iteration 341 logL: -13686.541015625
Iteration 342 logL: -13723.982421875
Iteration 343 logL: -13691.625
Iteration 344 logL: -13643.236328125
Iteration 345 logL: -13652.939453125
Iteration 346 logL: -13637.693359375
Iteration 347 logL: -13645.2841796875
Iteration 348 logL: -13639.3994140625
Iteration 349 logL: -13589.154296875
Iteration 350 logL: -13558.333984375
Iteration 351 logL: -13571.357421875
Iteration 352 logL: -13570.7763671875
Iteration 353 logL: -13563.939453125
Iteration 354 logL: -13499.6806640625
Iteration 355 logL: -13529.0576171875
Iteration 356 logL: -13499.02734375
Iteration 357 logL: -13464.775390625
Iteration 358 logL: -13461.4140625
Iteration 359 logL: -13439.3408203125
Iteration 360 logL: -13457.806640625
Iteration 361 logL: -13429.4609375
Iteration 362 logL: -13407.4013671875
Iteration 363 logL: -13420.458984375
Iteration 364 logL: -13361.3349609375
Iteration 365 logL: -13370.328125
Iteration 366 logL: -13349.671875
Iteration 367 logL: -13315.1796875
Iteration 368 logL: -13331.03515625
Iteration 369 logL: -13287.126953125
Iteration 370 logL: -13295.1953125
Iteration 371 logL: -13289.986328125
Iteration 372 logL: -13265.892578125
Iteration 373 logL: -13234.046875
Iteration 374 logL: -13232.55078125
Iteration 375 logL: -13227.5029296875
Iteration 376 logL: -13198.287109375
Iteration 377 logL: -13203.33984375
Iteration 378 logL: -13177.259765625
Iteration 379 logL: -13207.76953125
Iteration 380 logL: -13200.208984375
Iteration 381 logL: -13154.46875
Iteration 382 logL: -13100.6953125
Iteration 383 logL: -13092.6171875
Iteration 384 logL: -13080.6474609375
Iteration 385 logL: -13095.82421875
Iteration 386 logL: -13060.03125
Iteration 387 logL: -13081.5224609375
Iteration 388 logL: -13029.5029296875
Iteration 389 logL: -13000.671875
Iteration 390 logL: -13028.10546875
```

```
Iteration 391 logL: -12991.958984375
Iteration 392 logL: -12986.70703125
Iteration 393 logL: -12969.55078125
Iteration 394 logL: -12914.26953125
Iteration 395 logL: -12929.560546875
Iteration 396 logL: -12948.0146484375
Iteration 397 logL: -12921.8115234375
Iteration 398 logL: -12929.3818359375
Iteration 399 logL: -12870.3544921875
Iteration 400 logL: -12879.7373046875
Iteration 401 logL: -12888.7177734375
Iteration 402 logL: -12882.2373046875
Iteration 403 logL: -12822.31640625
Iteration 404 logL: -12800.3349609375
Iteration 405 logL: -12845.8671875
Iteration 406 logL: -12820.65234375
Iteration 407 logL: -12772.267578125
Iteration 408 logL: -12772.3681640625
Iteration 409 logL: -12727.240234375
Iteration 410 logL: -12727.609375
Iteration 411 logL: -12694.7607421875
Iteration 412 logL: -12709.646484375
Iteration 413 logL: -12707.3330078125
Iteration 414 logL: -12670.7265625
Iteration 415 logL: -12684.259765625
Iteration 416 logL: -12661.3857421875
Iteration 417 logL: -12647.03515625
Iteration 418 logL: -12644.1142578125
Iteration 419 logL: -12626.138671875
Iteration 420 logL: -12615.04296875
Iteration 421 logL: -12565.3623046875
Iteration 422 logL: -12585.7890625
Iteration 423 logL: -12535.427734375
Iteration 424 logL: -12588.8046875
Iteration 425 logL: -12558.8818359375
Iteration 426 logL: -12509.9521484375
Iteration 427 logL: -12496.037109375
Iteration 428 logL: -12517.509765625
Iteration 429 logL: -12479.228515625
Iteration 430 logL: -12462.359375
Iteration 431 logL: -12460.54296875
Iteration 432 logL: -12412.59375
Iteration 433 logL: -12452.04296875
Iteration 434 logL: -12422.56640625
Iteration 435 logL: -12421.1787109375
Iteration 436 logL: -12368.994140625
Iteration 437 logL: -12385.1943359375
Iteration 438 logL: -12326.5
Iteration 439 logL: -12366.515625
Iteration 440 logL: -12352.5
Iteration 441 logL: -12282.357421875
Iteration 442 logL: -12325.5703125
Iteration 443 logL: -12283.7763671875
Iteration 444 logL: -12261.728515625
Iteration 445 logL: -12294.96875
Iteration 446 logL: -12255.701171875
Iteration 447 logL: -12232.212890625
Iteration 448 logL: -12240.703125
Iteration 449 logL: -12199.4609375
```

```
Iteration 450 logL: -12230.5361328125
Iteration 451 logL: -12208.7119140625
Iteration 452 logL: -12175.228515625
Iteration 453 logL: -12133.08203125
Iteration 454 logL: -12151.5849609375
Iteration 455 logL: -12122.2265625
Iteration 456 logL: -12118.208984375
Iteration 457 logL: -12128.814453125
Iteration 458 logL: -12068.8720703125
Iteration 459 logL: -12024.6826171875
Iteration 460 logL: -12086.60546875
Iteration 461 logL: -12035.1630859375
Iteration 462 logL: -12037.7109375
Iteration 463 logL: -12021.326171875
Iteration 464 logL: -11985.955078125
Iteration 465 logL: -11957.1865234375
Iteration 466 logL: -11995.3720703125
Iteration 467 logL: -11987.763671875
Iteration 468 logL: -11944.2412109375
Iteration 469 logL: -11955.6025390625
Iteration 470 logL: -11907.923828125
Iteration 471 logL: -11899.19140625
Iteration 472 logL: -11902.8916015625
Iteration 473 logL: -11851.23828125
Iteration 474 logL: -11881.828125
Iteration 475 logL: -11845.001953125
Iteration 476 logL: -11847.9765625
Iteration 477 logL: -11839.4658203125
Iteration 478 logL: -11865.8076171875
Iteration 479 logL: -11792.619140625
Iteration 480 logL: -11792.583984375
Iteration 481 logL: -11756.107421875
Iteration 482 logL: -11794.072265625
Iteration 483 logL: -11759.0078125
Iteration 484 logL: -11708.3642578125
Iteration 485 logL: -11745.6328125
Iteration 486 logL: -11721.7900390625
Iteration 487 logL: -11693.2900390625
Iteration 488 logL: -11651.837890625
Iteration 489 logL: -11702.6884765625
Iteration 490 logL: -11654.45703125
Iteration 491 logL: -11640.984375
Iteration 492 logL: -11601.8974609375
Iteration 493 logL: -11629.23046875
Iteration 494 logL: -11602.34375
Iteration 495 logL: -11609.783203125
Iteration 496 logL: -11579.3115234375
Iteration 497 logL: -11542.3017578125
Iteration 498 logL: -11592.162109375
Iteration 499 logL: -11571.619140625
Iteration 500 logL: -11508.8828125
Iteration 501 logL: -11502.94921875
Iteration 502 logL: -11478.609375
Iteration 503 logL: -11506.30078125
Iteration 504 logL: -11504.892578125
Iteration 505 logL: -11475.0517578125
Iteration 506 logL: -11427.1220703125
Iteration 507 logL: -11434.119140625
Iteration 508 logL: -11394.5322265625
```

```
Iteration 509 logL: -11430.517578125
Iteration 510 logL: -11402.44140625
Iteration 511 logL: -11371.01953125
Iteration 512 logL: -11353.78515625
Iteration 513 logL: -11333.9892578125
Iteration 514 logL: -11290.52734375
Iteration 515 logL: -11346.5751953125
Iteration 516 logL: -11312.45703125
Iteration 517 logL: -11262.630859375
Iteration 518 logL: -11305.216796875
Iteration 519 logL: -11255.810546875
Iteration 520 logL: -11224.05078125
Iteration 521 logL: -11252.6162109375
Iteration 522 logL: -11239.716796875
Iteration 523 logL: -11242.396484375
Iteration 524 logL: -11201.576171875
Iteration 525 logL: -11195.59765625
Iteration 526 logL: -11192.0361328125
Iteration 527 logL: -11153.4453125
Iteration 528 logL: -11142.826171875
Iteration 529 logL: -11125.490234375
Iteration 530 logL: -11121.55078125
Iteration 531 logL: -11132.361328125
Iteration 532 logL: -11073.4599609375
Iteration 533 logL: -11095.46484375
Iteration 534 logL: -11063.0703125
Iteration 535 logL: -11039.3642578125
Iteration 536 logL: -11064.798828125
Iteration 537 logL: -10993.8251953125
Iteration 538 logL: -11016.6796875
Iteration 539 logL: -11024.9765625
Iteration 540 logL: -10999.3828125
Iteration 541 logL: -10956.189453125
Iteration 542 logL: -10996.0634765625
Iteration 543 logL: -10940.951171875
Iteration 544 logL: -10955.734375
Iteration 545 logL: -10877.4853515625
Iteration 546 logL: -10909.06640625
Iteration 547 logL: -10855.9697265625
Iteration 548 logL: -10892.7470703125
Iteration 549 logL: -10860.56640625
Iteration 550 logL: -10862.7001953125
Iteration 551 logL: -10828.95703125
Iteration 552 logL: -10817.82421875
Iteration 553 logL: -10834.794921875
Iteration 554 logL: -10794.822265625
Iteration 555 logL: -10801.373046875
Iteration 556 logL: -10763.212890625
Iteration 557 logL: -10783.5361328125
Iteration 558 logL: -10700.751953125
Iteration 559 logL: -10694.4072265625
Iteration 560 logL: -10740.7529296875
Iteration 561 logL: -10708.08984375
Iteration 562 logL: -10703.017578125
Iteration 563 logL: -10703.154296875
Iteration 564 logL: -10689.9375
Iteration 565 logL: -10663.6396484375
Iteration 566 logL: -10667.341796875
Iteration 567 logL: -10647.71875
```

```
Iteration 568 logL: -10627.951171875
Iteration 569 logL: -10612.609375
Iteration 570 logL: -10555.623046875
Iteration 571 logL: -10582.03125
Iteration 572 logL: -10549.6005859375
Iteration 573 logL: -10507.9189453125
Iteration 574 logL: -10516.3203125
Iteration 575 logL: -10513.064453125
Iteration 576 logL: -10536.09375
Iteration 577 logL: -10514.9951171875
Iteration 578 logL: -10475.8076171875
Iteration 579 logL: -10491.5615234375
Iteration 580 logL: -10434.8095703125
Iteration 581 logL: -10428.953125
Iteration 582 logL: -10424.9423828125
Iteration 583 logL: -10404.70703125
Iteration 584 logL: -10373.0654296875
Iteration 585 logL: -10386.2724609375
Iteration 586 logL: -10315.74609375
Iteration 587 logL: -10332.6962890625
Iteration 588 logL: -10327.48046875
Iteration 589 logL: -10305.947265625
Iteration 590 logL: -10332.01171875
Iteration 591 logL: -10273.9892578125
Iteration 592 logL: -10294.041015625
Iteration 593 logL: -10237.146484375
Iteration 594 logL: -10278.087890625
Iteration 595 logL: -10239.181640625
Iteration 596 logL: -10246.458984375
Iteration 597 logL: -10217.6943359375
Iteration 598 logL: -10227.6494140625
Iteration 599 logL: -10223.2001953125
Iteration 600 logL: -10168.8515625
Iteration 601 logL: -10129.9541015625
Iteration 602 logL: -10179.0771484375
Iteration 603 logL: -10139.896484375
Iteration 604 logL: -10116.0107421875
Iteration 605 logL: -10128.21484375
Iteration 606 logL: -10079.7421875
Iteration 607 logL: -10081.2802734375
Iteration 608 logL: -10072.419921875
Iteration 609 logL: -10054.6171875
Iteration 610 logL: -10003.2890625
Iteration 611 logL: -9988.310546875
Iteration 612 logL: -10015.2138671875
Iteration 613 logL: -9984.119140625
Iteration 614 logL: -9968.505859375
Iteration 615 logL: -9977.9375
Iteration 616 logL: -9939.8427734375
Iteration 617 logL: -9991.857421875
Iteration 618 logL: -9962.375
Iteration 619 logL: -9918.2607421875
Iteration 620 logL: -9921.2841796875
Iteration 621 logL: -9918.177734375
Iteration 622 logL: -9891.302734375
Iteration 623 logL: -9898.2109375
Iteration 624 logL: -9815.314453125
Iteration 625 logL: -9836.0595703125
Iteration 626 logL: -9865.3583984375
```

```
Iteration 627 logL: -9756.5029296875
Iteration 628 logL: -9764.224609375
Iteration 629 logL: -9733.7890625
Iteration 630 logL: -9755.650390625
Iteration 631 logL: -9689.388671875
Iteration 632 logL: -9783.068359375
Iteration 633 logL: -9682.283203125
Iteration 634 logL: -9714.6630859375
Iteration 635 logL: -9733.58203125
Iteration 636 logL: -9708.2001953125
Iteration 637 logL: -9687.3359375
Iteration 638 logL: -9706.2587890625
Iteration 639 logL: -9779.482421875
Iteration 640 logL: -9642.357421875
Iteration 641 logL: -9660.1123046875
Iteration 642 logL: -9646.74609375
Iteration 643 logL: -9589.115234375
Iteration 644 logL: -9621.4482421875
Iteration 645 logL: -9568.0947265625
Iteration 646 logL: -9567.658203125
Iteration 647 logL: -9538.1962890625
Iteration 648 logL: -9524.53125
Iteration 649 logL: -9515.146484375
Iteration 650 logL: -9477.513671875
Iteration 651 logL: -9463.25
Iteration 652 logL: -9491.9541015625
Iteration 653 logL: -9436.9853515625
Iteration 654 logL: -9419.7392578125
Iteration 655 logL: -9447.1005859375
Iteration 656 logL: -9413.6640625
Iteration 657 logL: -9399.037109375
Iteration 658 logL: -9376.6484375
Iteration 659 logL: -9368.125
Iteration 660 logL: -9394.6865234375
Iteration 661 logL: -9361.73046875
Iteration 662 logL: -9343.4052734375
Iteration 663 logL: -9357.5703125
Iteration 664 logL: -9317.212890625
Iteration 665 logL: -9281.421875
Iteration 666 logL: -9294.4912109375
Iteration 667 logL: -9313.1650390625
Iteration 668 logL: -9270.7177734375
Iteration 669 logL: -9208.640625
Iteration 670 logL: -9259.134765625
Iteration 671 logL: -9222.314453125
Iteration 672 logL: -9220.609375
Iteration 673 logL: -9179.2861328125
Iteration 674 logL: -9196.2099609375
Iteration 675 logL: -9156.9462890625
Iteration 676 logL: -9145.26171875
Iteration 677 logL: -9209.81640625
Iteration 678 logL: -9178.5380859375
Iteration 679 logL: -9094.908203125
Iteration 680 logL: -9131.052734375
Iteration 681 logL: -9160.609375
Iteration 682 logL: -9112.94921875
Iteration 683 logL: -9121.34375
Iteration 684 logL: -9083.6015625
Iteration 685 logL: -9035.9833984375
```

```
Iteration 686 logL: -9029.7548828125
Iteration 687 logL: -9050.2734375
Iteration 688 logL: -9002.6201171875
Iteration 689 logL: -8982.1328125
Iteration 690 logL: -8963.421875
Iteration 691 logL: -8984.2265625
Iteration 692 logL: -8943.1953125
Iteration 693 logL: -8918.7822265625
Iteration 694 logL: -8964.2978515625
Iteration 695 logL: -8906.81640625
Iteration 696 logL: -8910.5419921875
Iteration 697 logL: -8926.953125
Iteration 698 logL: -8921.685546875
Iteration 699 logL: -8878.154296875
Iteration 700 logL: -8839.798828125
Iteration 701 logL: -8859.958984375
Iteration 702 logL: -8786.5087890625
Iteration 703 logL: -8824.2568359375
Iteration 704 logL: -8824.8349609375
Iteration 705 logL: -8788.0986328125
Iteration 706 logL: -8800.716796875
Iteration 707 logL: -8746.3515625
Iteration 708 logL: -8755.919921875
Iteration 709 logL: -8762.5986328125
Iteration 710 logL: -8706.1494140625
Iteration 711 logL: -8727.3046875
Iteration 712 logL: -8653.17578125
Iteration 713 logL: -8668.775390625
Iteration 714 logL: -8680.857421875
Iteration 715 logL: -8690.7734375
Iteration 716 logL: -8638.8505859375
Iteration 717 logL: -8623.494140625
Iteration 718 logL: -8626.19921875
Iteration 719 logL: -8558.3671875
Iteration 720 logL: -8604.634765625
Iteration 721 logL: -8612.78125
Iteration 722 logL: -8523.640625
Iteration 723 logL: -8531.16015625
Iteration 724 logL: -8601.5166015625
Iteration 725 logL: -8471.853515625
Iteration 726 logL: -8493.767578125
Iteration 727 logL: -8475.365234375
Iteration 728 logL: -8539.34765625
Iteration 729 logL: -8488.150390625
Iteration 730 logL: -8433.6796875
Iteration 731 logL: -8467.142578125
Iteration 732 logL: -8437.9404296875
Iteration 733 logL: -8422.39453125
Iteration 734 logL: -8361.837890625
Iteration 735 logL: -8469.3212890625
Iteration 736 logL: -8342.935546875
Iteration 737 logL: -8343.53125
Iteration 738 logL: -8295.1279296875
Iteration 739 logL: -8337.2763671875
Iteration 740 logL: -8306.3642578125
Iteration 741 logL: -8271.2109375
Iteration 742 logL: -8271.75390625
Iteration 743 logL: -8241.068359375
Iteration 744 logL: -8221.61328125
```

```
Iteration 745 logL: -8188.70947265625
Iteration 746 logL: -8239.107421875
Iteration 747 logL: -8187.53125
Iteration 748 logL: -8164.658203125
Iteration 749 logL: -8139.17822265625
Iteration 750 logL: -8192.0419921875
Iteration 751 logL: -8071.515625
Iteration 752 logL: -8201.7958984375
Iteration 753 logL: -8111.5087890625
Iteration 754 logL: -8134.3173828125
Iteration 755 logL: -8073.03125
Iteration 756 logL: -8048.5400390625
Iteration 757 logL: -8068.103515625
Iteration 758 logL: -8055.40625
Iteration 759 logL: -7978.666015625
Iteration 760 logL: -8036.0
Iteration 761 logL: -7969.6396484375
Iteration 762 logL: -7964.5634765625
Iteration 763 logL: -7946.99267578125
Iteration 764 logL: -7950.478515625
Iteration 765 logL: -7891.880859375
Iteration 766 logL: -7907.69873046875
Iteration 767 logL: -7957.05322265625
Iteration 768 logL: -7836.63232421875
Iteration 769 logL: -7930.71728515625
Iteration 770 logL: -7836.96630859375
Iteration 771 logL: -7810.02685546875
Iteration 772 logL: -7885.12646484375
Iteration 773 logL: -7862.35009765625
Iteration 774 logL: -7744.42431640625
Iteration 775 logL: -7866.4716796875
Iteration 776 logL: -7834.2548828125
Iteration 777 logL: -7691.76611328125
Iteration 778 logL: -7773.9501953125
Iteration 779 logL: -7771.55810546875
Iteration 780 logL: -7766.4716796875
Iteration 781 logL: -7735.197265625
Iteration 782 logL: -7626.1591796875
Iteration 783 logL: -7755.8466796875
Iteration 784 logL: -7614.72412109375
Iteration 785 logL: -7574.087890625
Iteration 786 logL: -7600.1591796875
Iteration 787 logL: -7636.025390625
Iteration 788 logL: -7542.7373046875
Iteration 789 logL: -7521.84375
Iteration 790 logL: -7603.2197265625
Iteration 791 logL: -7561.4267578125
Iteration 792 logL: -7509.86474609375
Iteration 793 logL: -7461.5419921875
Iteration 794 logL: -7529.482421875
Iteration 795 logL: -7447.4033203125
Iteration 796 logL: -7410.080078125
Iteration 797 logL: -7456.56298828125
Iteration 798 logL: -7369.05615234375
Iteration 799 logL: -7465.28466796875
Iteration 800 logL: -7381.0849609375
Iteration 801 logL: -7495.50048828125
Iteration 802 logL: -7467.953125
Iteration 803 logL: -7350.7138671875
```

```
Iteration 804 logL: -7273.73095703125
Iteration 805 logL: -8084.58251953125
Iteration 806 logL: -7237.16748046875
Iteration 807 logL: -7544.828125
Iteration 808 logL: -7457.611328125
Iteration 809 logL: -7298.30859375
Iteration 810 logL: -7256.14453125
Iteration 811 logL: -7418.9169921875
Iteration 812 logL: -7136.7734375
Iteration 813 logL: -7278.841796875
Iteration 814 logL: -7332.6083984375
Iteration 815 logL: -7325.67919921875
Iteration 816 logL: -7132.04345703125
Iteration 817 logL: -7285.85546875
Iteration 818 logL: -7216.5009765625
Iteration 819 logL: -7128.44921875
Iteration 820 logL: -7097.2568359375
Iteration 821 logL: -7130.74755859375
Iteration 822 logL: -7192.951171875
Iteration 823 logL: -7109.15380859375
Iteration 824 logL: -7095.8076171875
Iteration 825 logL: -7066.76953125
Iteration 826 logL: -6934.7080078125
Iteration 827 logL: -6959.1142578125
Iteration 828 logL: -6942.7626953125
Iteration 829 logL: -6937.025390625
Iteration 830 logL: -7138.7216796875
Iteration 831 logL: -7503.6728515625
Iteration 832 logL: -6873.77734375
Iteration 833 logL: -7072.93017578125
Iteration 834 logL: -7290.123046875
Iteration 835 logL: -7176.7529296875
Iteration 836 logL: -6790.55859375
Iteration 837 logL: -6894.95166015625
Iteration 838 logL: -7084.5390625
Iteration 839 logL: -7109.8837890625
Iteration 840 logL: -6832.216796875
Iteration 841 logL: -6867.46484375
Iteration 842 logL: -6893.484375
Iteration 843 logL: -6700.130859375
Iteration 844 logL: -6767.677734375
Iteration 845 logL: -6769.08984375
Iteration 846 logL: -6627.248046875
Iteration 847 logL: -6747.0546875
Iteration 848 logL: -6674.31201171875
Iteration 849 logL: -6828.298828125
Iteration 850 logL: -6738.11181640625
Iteration 851 logL: -6817.3095703125
Iteration 852 logL: -6709.8828125
Iteration 853 logL: -6802.15087890625
Iteration 854 logL: -6807.88671875
Iteration 855 logL: -6888.70263671875
Iteration 856 logL: -6592.5576171875
Iteration 857 logL: -6586.490234375
Iteration 858 logL: -6773.0888671875
Iteration 859 logL: -6627.2119140625
Iteration 860 logL: -6633.7626953125
Iteration 861 logL: -6613.7587890625
Iteration 862 logL: -6600.3408203125
```

```
Iteration 863 logL: -6602.0595703125
Iteration 864 logL: -6641.068359375
Iteration 865 logL: -6507.02978515625
Iteration 866 logL: -6466.43603515625
Iteration 867 logL: -6658.087890625
Iteration 868 logL: -6478.8662109375
Iteration 869 logL: -6388.6435546875
Iteration 870 logL: -6382.322265625
Iteration 871 logL: -6392.30224609375
Iteration 872 logL: -6265.9658203125
Iteration 873 logL: -6340.640625
Iteration 874 logL: -6363.34765625
Iteration 875 logL: -6349.79296875
Iteration 876 logL: -6406.12353515625
Iteration 877 logL: -6457.5595703125
Iteration 878 logL: -6255.2470703125
Iteration 879 logL: -6496.4970703125
Iteration 880 logL: -6310.33203125
Iteration 881 logL: -6182.740234375
Iteration 882 logL: -6108.1318359375
Iteration 883 logL: -6253.62353515625
Iteration 884 logL: -6080.7578125
Iteration 885 logL: -6143.8984375
Iteration 886 logL: -6169.0283203125
Iteration 887 logL: -6206.8310546875
Iteration 888 logL: -6286.1318359375
Iteration 889 logL: -6105.7060546875
Iteration 890 logL: -6220.1357421875
Iteration 891 logL: -6091.55078125
Iteration 892 logL: -6136.7939453125
Iteration 893 logL: -6095.728515625
Iteration 894 logL: -5973.1923828125
Iteration 895 logL: -5949.7998046875
Iteration 896 logL: -6132.755859375
Iteration 897 logL: -6044.1689453125
Iteration 898 logL: -5895.63623046875
Iteration 899 logL: -6002.69970703125
Iteration 900 logL: -6025.7265625
Iteration 901 logL: -6125.97998046875
Iteration 902 logL: -6044.61328125
Iteration 903 logL: -5929.60986328125
Iteration 904 logL: -5962.2998046875
Iteration 905 logL: -6055.421875
Iteration 906 logL: -6345.9287109375
Iteration 907 logL: -6035.169921875
Iteration 908 logL: -5919.2724609375
Iteration 909 logL: -5894.68408203125
Iteration 910 logL: -5944.14453125
Iteration 911 logL: -5822.15234375
Iteration 912 logL: -6138.86572265625
Iteration 913 logL: -5811.5712890625
Iteration 914 logL: -6203.0830078125
Iteration 915 logL: -6201.388671875
Iteration 916 logL: -5840.1748046875
Iteration 917 logL: -5772.6005859375
Iteration 918 logL: -6065.7646484375
Iteration 919 logL: -6123.12158203125
Iteration 920 logL: -5810.6298828125
Iteration 921 logL: -5651.958984375
```

```
Iteration 922 logL: -5878.81689453125
Iteration 923 logL: -6384.916015625
Iteration 924 logL: -5783.3564453125
Iteration 925 logL: -5630.8798828125
Iteration 926 logL: -5804.8818359375
Iteration 927 logL: -5720.53955078125
Iteration 928 logL: -6159.7958984375
Iteration 929 logL: -5737.958984375
Iteration 930 logL: -5582.4921875
Iteration 931 logL: -5546.17333984375
Iteration 932 logL: -5753.5771484375
Iteration 933 logL: -5701.69921875
Iteration 934 logL: -5619.6630859375
Iteration 935 logL: -5631.33203125
Iteration 936 logL: -5472.2021484375
Iteration 937 logL: -5567.48828125
Iteration 938 logL: -5637.1376953125
Iteration 939 logL: -5596.400390625
Iteration 940 logL: -5696.048828125
Iteration 941 logL: -5428.2578125
Iteration 942 logL: -5435.07177734375
Iteration 943 logL: -5649.953125
Iteration 944 logL: -5803.3134765625
Iteration 945 logL: -5342.697265625
Iteration 946 logL: -5489.1103515625
Iteration 947 logL: -5286.79736328125
Iteration 948 logL: -5433.88525390625
Iteration 949 logL: -5474.21484375
Iteration 950 logL: -5503.6787109375
Iteration 951 logL: -5564.46630859375
Iteration 952 logL: -5372.56201171875
Iteration 953 logL: -5285.521484375
Iteration 954 logL: -5434.37890625
Iteration 955 logL: -5526.681640625
Iteration 956 logL: -5499.0498046875
Iteration 957 logL: -5232.873046875
Iteration 958 logL: -5285.06298828125
Iteration 959 logL: -5297.06689453125
Iteration 960 logL: -5507.53466796875
Iteration 961 logL: -5319.330078125
Iteration 962 logL: -5137.255859375
Iteration 963 logL: -5404.34814453125
Iteration 964 logL: -5156.2021484375
Iteration 965 logL: -5067.1533203125
Iteration 966 logL: -5419.8759765625
Iteration 967 logL: -5284.595703125
Iteration 968 logL: -5114.994140625
Iteration 969 logL: -5107.8662109375
Iteration 970 logL: -5045.2041015625
Iteration 971 logL: -4924.28466796875
Iteration 972 logL: -4961.044921875
Iteration 973 logL: -5088.001953125
Iteration 974 logL: -4968.68017578125
Iteration 975 logL: -4924.9462890625
Iteration 976 logL: -4898.9384765625
Iteration 977 logL: -5048.2548828125
Iteration 978 logL: -5025.77880859375
Iteration 979 logL: -4756.1650390625
Iteration 980 logL: -4786.67578125
```

```
Iteration 981 logL: -5014.28125
Iteration 982 logL: -4881.173828125
Iteration 983 logL: -4967.2880859375
Iteration 984 logL: -4760.8271484375
Iteration 985 logL: -4737.5439453125
Iteration 986 logL: -4843.4716796875
Iteration 987 logL: -4685.263671875
Iteration 988 logL: -4719.29541015625
Iteration 989 logL: -4741.88525390625
Iteration 990 logL: -4791.91015625
Iteration 991 logL: -4678.41015625
Iteration 992 logL: -4687.28955078125
Iteration 993 logL: -4803.90234375
Iteration 994 logL: -4764.78125
Iteration 995 logL: -4550.07861328125
Iteration 996 logL: -4606.3837890625
Iteration 997 logL: -4655.45703125
Iteration 998 logL: -4631.99951171875
Iteration 999 logL: -4435.37255859375
Iteration 1000 logL: -4361.21337890625
Iteration 1001 logL: -4599.857421875
Iteration 1002 logL: -4757.986328125
Iteration 1003 logL: -4391.6904296875
Iteration 1004 logL: -4550.43017578125
Iteration 1005 logL: -4967.2919921875
Iteration 1006 logL: -4544.5283203125
Iteration 1007 logL: -4652.2470703125
Iteration 1008 logL: -4758.7138671875
Iteration 1009 logL: -4489.74267578125
Iteration 1010 logL: -4516.1591796875
Iteration 1011 logL: -4519.2880859375
Iteration 1012 logL: -4393.939453125
Iteration 1013 logL: -4975.9541015625
Iteration 1014 logL: -4429.96484375
Iteration 1015 logL: -4571.923828125
Iteration 1016 logL: -4728.4873046875
Iteration 1017 logL: -4301.2607421875
Iteration 1018 logL: -5350.1435546875
Iteration 1019 logL: -4811.67529296875
Iteration 1020 logL: -4452.41796875
Iteration 1021 logL: -4163.38232421875
Iteration 1022 logL: -4310.0048828125
Iteration 1023 logL: -5092.3818359375
Iteration 1024 logL: -4989.4736328125
Iteration 1025 logL: -4412.96240234375
Iteration 1026 logL: -4709.4130859375
Iteration 1027 logL: -4903.78662109375
Iteration 1028 logL: -4942.1162109375
Iteration 1029 logL: -4644.24560546875
Iteration 1030 logL: -4320.79638671875
Iteration 1031 logL: -4738.05517578125
Iteration 1032 logL: -4887.06298828125
Iteration 1033 logL: -4461.46728515625
Iteration 1034 logL: -4203.02001953125
Iteration 1035 logL: -4154.072265625
Iteration 1036 logL: -4293.73974609375
Iteration 1037 logL: -4408.07470703125
Iteration 1038 logL: -4334.740234375
Iteration 1039 logL: -4237.8935546875
```

```
Iteration 1040 logL: -4116.56787109375
Iteration 1041 logL: -4158.826171875
Iteration 1042 logL: -4306.8046875
Iteration 1043 logL: -4029.239990234375
Iteration 1044 logL: -4224.74755859375
Iteration 1045 logL: -4157.94921875
Iteration 1046 logL: -4009.492431640625
Iteration 1047 logL: -4009.404541015625
Iteration 1048 logL: -4165.5390625
Iteration 1049 logL: -4152.9892578125
Iteration 1050 logL: -4055.03125
Iteration 1051 logL: -4019.0673828125
Iteration 1052 logL: -4207.72412109375
Iteration 1053 logL: -4243.7724609375
Iteration 1054 logL: -4216.3583984375
Iteration 1055 logL: -3962.40234375
Iteration 1056 logL: -4190.505859375
Iteration 1057 logL: -3987.20068359375
Iteration 1058 logL: -4181.720703125
Iteration 1059 logL: -4248.052734375
Iteration 1060 logL: -4448.80078125
Iteration 1061 logL: -4196.05810546875
Iteration 1062 logL: -3996.36962890625
Iteration 1063 logL: -3989.151123046875
Iteration 1064 logL: -4271.7744140625
Iteration 1065 logL: -4408.59033203125
Iteration 1066 logL: -4031.7373046875
Iteration 1067 logL: -4127.57470703125
Iteration 1068 logL: -3894.702880859375
Iteration 1069 logL: -4227.20751953125
Iteration 1070 logL: -4221.73388671875
Iteration 1071 logL: -3973.160888671875
Iteration 1072 logL: -3887.02978515625
Iteration 1073 logL: -3824.8037109375
Iteration 1074 logL: -3902.9404296875
Iteration 1075 logL: -3806.489501953125
Iteration 1076 logL: -3865.654052734375
Iteration 1077 logL: -4042.7763671875
Iteration 1078 logL: -3876.6748046875
Iteration 1079 logL: -3720.94873046875
Iteration 1080 logL: -3834.37939453125
Iteration 1081 logL: -3806.0576171875
Iteration 1082 logL: -4041.0888671875
Iteration 1083 logL: -3675.787353515625
Iteration 1084 logL: -3642.779541015625
Iteration 1085 logL: -3818.5087890625
Iteration 1086 logL: -3899.248779296875
Iteration 1087 logL: -3822.533935546875
Iteration 1088 logL: -3926.136474609375
Iteration 1089 logL: -3904.84765625
Iteration 1090 logL: -3713.929443359375
Iteration 1091 logL: -3956.45068359375
Iteration 1092 logL: -3858.43701171875
Iteration 1093 logL: -3693.076171875
Iteration 1094 logL: -3509.55712890625
Iteration 1095 logL: -3695.112060546875
Iteration 1096 logL: -3709.49365234375
Iteration 1097 logL: -3508.514892578125
Iteration 1098 logL: -3675.20166015625
```

```
Iteration 1099 logL: -3576.557373046875
Iteration 1100 logL: -4130.212890625
Iteration 1101 logL: -3660.2744140625
Iteration 1102 logL: -3415.10546875
Iteration 1103 logL: -3508.38427734375
Iteration 1104 logL: -3898.27880859375
Iteration 1105 logL: -3842.91015625
Iteration 1106 logL: -3481.013671875
Iteration 1107 logL: -3530.695556640625
Iteration 1108 logL: -3451.71875
Iteration 1109 logL: -3416.2529296875
Iteration 1110 logL: -3334.884521484375
Iteration 1111 logL: -3614.9580078125
Iteration 1112 logL: -3404.63427734375
Iteration 1113 logL: -3345.651123046875
Iteration 1114 logL: -3351.63525390625
Iteration 1115 logL: -3349.2763671875
Iteration 1116 logL: -3474.748046875
Iteration 1117 logL: -3288.562744140625
Iteration 1118 logL: -3463.14892578125
Iteration 1119 logL: -3436.834228515625
Iteration 1120 logL: -3411.611572265625
Iteration 1121 logL: -3277.4609375
Iteration 1122 logL: -3397.420654296875
Iteration 1123 logL: -3211.050048828125
Iteration 1124 logL: -3383.0078125
Iteration 1125 logL: -3363.18115234375
Iteration 1126 logL: -3279.631103515625
Iteration 1127 logL: -3661.919677734375
Iteration 1128 logL: -3359.66796875
Iteration 1129 logL: -3269.910400390625
Iteration 1130 logL: -3196.595947265625
Iteration 1131 logL: -3404.35791015625
Iteration 1132 logL: -3286.587158203125
Iteration 1133 logL: -3175.717041015625
Iteration 1134 logL: -3751.96240234375
Iteration 1135 logL: -3222.361328125
Iteration 1136 logL: -3090.076171875
Iteration 1137 logL: -3323.45068359375
Iteration 1138 logL: -3384.003173828125
Iteration 1139 logL: -3106.176025390625
Iteration 1140 logL: -3244.28466796875
Iteration 1141 logL: -3295.09765625
Iteration 1142 logL: -3373.354736328125
Iteration 1143 logL: -3604.62158203125
Iteration 1144 logL: -2997.947021484375
Iteration 1145 logL: -3138.3828125
Iteration 1146 logL: -3470.816650390625
Iteration 1147 logL: -3404.827880859375
Iteration 1148 logL: -3152.863525390625
Iteration 1149 logL: -3132.96142578125
Iteration 1150 logL: -3394.03564453125
Iteration 1151 logL: -3419.419677734375
Iteration 1152 logL: -3118.311279296875
Iteration 1153 logL: -3702.74951171875
Iteration 1154 logL: -3494.944091796875
Iteration 1155 logL: -2969.537841796875
Iteration 1156 logL: -3136.345703125
Iteration 1157 logL: -3167.41162109375
```

```
Iteration 1158 logL: -3357.986083984375
Iteration 1159 logL: -3184.684326171875
Iteration 1160 logL: -3088.1298828125
Iteration 1161 logL: -2873.75634765625
Iteration 1162 logL: -3251.76123046875
Iteration 1163 logL: -3068.23291015625
Iteration 1164 logL: -3066.56103515625
Iteration 1165 logL: -2956.0029296875
Iteration 1166 logL: -2994.2666015625
Iteration 1167 logL: -3225.078369140625
Iteration 1168 logL: -3009.1845703125
Iteration 1169 logL: -2998.830078125
Iteration 1170 logL: -2940.505859375
Iteration 1171 logL: -2990.271240234375
Iteration 1172 logL: -3473.5244140625
Iteration 1173 logL: -3115.8515625
Iteration 1174 logL: -3021.552001953125
Iteration 1175 logL: -3080.023681640625
Iteration 1176 logL: -3098.281494140625
Iteration 1177 logL: -2848.409912109375
Iteration 1178 logL: -2859.49462890625
Iteration 1179 logL: -3028.299072265625
Iteration 1180 logL: -2962.333740234375
Iteration 1181 logL: -2986.6357421875
Iteration 1182 logL: -2745.250244140625
Iteration 1183 logL: -2683.600341796875
Iteration 1184 logL: -2913.44873046875
Iteration 1185 logL: -3005.700927734375
Iteration 1186 logL: -3196.783447265625
Iteration 1187 logL: -3054.088623046875
Iteration 1188 logL: -2869.123779296875
Iteration 1189 logL: -3245.860107421875
Iteration 1190 logL: -2892.712646484375
Iteration 1191 logL: -2884.71533203125
Iteration 1192 logL: -3065.6376953125
Iteration 1193 logL: -3081.58544921875
Iteration 1194 logL: -2825.65673828125
Iteration 1195 logL: -3053.335205078125
Iteration 1196 logL: -2825.814697265625
Iteration 1197 logL: -2684.968505859375
Iteration 1198 logL: -3195.7705078125
Iteration 1199 logL: -2653.782958984375
Iteration 1200 logL: -2881.01513671875
Iteration 1201 logL: -2594.373046875
Iteration 1202 logL: -3186.290771484375
Iteration 1203 logL: -2669.77001953125
Iteration 1204 logL: -2681.761962890625
Iteration 1205 logL: -2778.9091796875
Iteration 1206 logL: -2771.330810546875
Iteration 1207 logL: -3045.22509765625
Iteration 1208 logL: -2528.1962890625
Iteration 1209 logL: -2505.445068359375
Iteration 1210 logL: -2668.7626953125
Iteration 1211 logL: -2642.599853515625
Iteration 1212 logL: -2868.640380859375
Iteration 1213 logL: -2564.613037109375
Iteration 1214 logL: -2587.291259765625
Iteration 1215 logL: -2991.3310546875
Iteration 1216 logL: -2990.892578125
```

```
Iteration 1217 logL: -2573.41357421875
Iteration 1218 logL: -2680.608154296875
Iteration 1219 logL: -2518.35302734375
Iteration 1220 logL: -3112.60498046875
Iteration 1221 logL: -3123.86572265625
Iteration 1222 logL: -2724.184814453125
Iteration 1223 logL: -2557.393310546875
Iteration 1224 logL: -2791.5908203125
Iteration 1225 logL: -2433.333251953125
Iteration 1226 logL: -2912.706298828125
Iteration 1227 logL: -2720.118896484375
Iteration 1228 logL: -2365.4248046875
Iteration 1229 logL: -2536.8916015625
Iteration 1230 logL: -2597.728759765625
Iteration 1231 logL: -2620.397216796875
Iteration 1232 logL: -3119.343505859375
Iteration 1233 logL: -2704.538818359375
Iteration 1234 logL: -2825.074462890625
Iteration 1235 logL: -2842.80712890625
Iteration 1236 logL: -2599.98779296875
Iteration 1237 logL: -2385.271484375
Iteration 1238 logL: -2523.34814453125
Iteration 1239 logL: -2555.826904296875
Iteration 1240 logL: -2489.4208984375
Iteration 1241 logL: -2637.805908203125
Iteration 1242 logL: -2376.208984375
Iteration 1243 logL: -2416.0732421875
Iteration 1244 logL: -2779.931640625
Iteration 1245 logL: -2476.144287109375
Iteration 1246 logL: -2508.439208984375
Iteration 1247 logL: -2821.218994140625
Iteration 1248 logL: -2250.033935546875
Iteration 1249 logL: -3003.495849609375
Iteration 1250 logL: -2359.261962890625
Iteration 1251 logL: -2547.18505859375
Iteration 1252 logL: -2528.8505859375
Iteration 1253 logL: -2630.987060546875
Iteration 1254 logL: -2542.70556640625
Iteration 1255 logL: -2649.642578125
Iteration 1256 logL: -2706.236328125
Iteration 1257 logL: -2255.3623046875
Iteration 1258 logL: -2462.779052734375
Iteration 1259 logL: -2408.765380859375
Iteration 1260 logL: -2431.61865234375
Iteration 1261 logL: -2578.74365234375
Iteration 1262 logL: -2339.771728515625
Iteration 1263 logL: -2373.67626953125
Iteration 1264 logL: -2914.001953125
Iteration 1265 logL: -2697.147705078125
Iteration 1266 logL: -2364.197021484375
Iteration 1267 logL: -2254.84033203125
Iteration 1268 logL: -2482.595458984375
Iteration 1269 logL: -2362.504638671875
Iteration 1270 logL: -2476.61962890625
Iteration 1271 logL: -2158.739013671875
Iteration 1272 logL: -2202.883544921875
Iteration 1273 logL: -2221.578857421875
Iteration 1274 logL: -2235.981201171875
Iteration 1275 logL: -2337.4052734375
```

```
Iteration 1276 logL: -2310.697509765625
Iteration 1277 logL: -2376.33154296875
Iteration 1278 logL: -2203.706787109375
Iteration 1279 logL: -2352.3720703125
Iteration 1280 logL: -2237.9541015625
Iteration 1281 logL: -2394.78857421875
Iteration 1282 logL: -2343.3349609375
Iteration 1283 logL: -2094.46826171875
Iteration 1284 logL: -2216.79296875
Iteration 1285 logL: -2404.307373046875
Iteration 1286 logL: -2081.377685546875
Iteration 1287 logL: -2507.8740234375
Iteration 1288 logL: -2229.82568359375
Iteration 1289 logL: -2205.24072265625
Iteration 1290 logL: -2184.66552734375
Iteration 1291 logL: -2052.580078125
Iteration 1292 logL: -1996.965576171875
Iteration 1293 logL: -2213.884521484375
Iteration 1294 logL: -2263.972412109375
Iteration 1295 logL: -2260.785400390625
Iteration 1296 logL: -2083.482421875
Iteration 1297 logL: -2264.184814453125
Iteration 1298 logL: -2069.980712890625
Iteration 1299 logL: -2263.10888671875
Iteration 1300 logL: -2107.623046875
Iteration 1301 logL: -2171.81640625
Iteration 1302 logL: -2559.289794921875
Iteration 1303 logL: -1943.68115234375
Iteration 1304 logL: -2152.36279296875
Iteration 1305 logL: -2334.0869140625
Iteration 1306 logL: -1985.1239013671875
Iteration 1307 logL: -2255.921630859375
Iteration 1308 logL: -2206.647216796875
Iteration 1309 logL: -2155.489013671875
Iteration 1310 logL: -2141.217529296875
Iteration 1311 logL: -2005.884033203125
Iteration 1312 logL: -2068.6787109375
Iteration 1313 logL: -2141.31884765625
Iteration 1314 logL: -2141.726806640625
Iteration 1315 logL: -2270.521484375
Iteration 1316 logL: -2157.947265625
Iteration 1317 logL: -1904.255615234375
Iteration 1318 logL: -2106.03857421875
Iteration 1319 logL: -2391.640625
Iteration 1320 logL: -2068.9599609375
Iteration 1321 logL: -2022.915771484375
Iteration 1322 logL: -2372.29541015625
Iteration 1323 logL: -2280.930419921875
Iteration 1324 logL: -2435.249267578125
Iteration 1325 logL: -1866.242431640625
Iteration 1326 logL: -1939.09765625
Iteration 1327 logL: -2532.3876953125
Iteration 1328 logL: -2327.2529296875
Iteration 1329 logL: -1980.422607421875
Iteration 1330 logL: -2005.8568115234375
Iteration 1331 logL: -2215.61279296875
Iteration 1332 logL: -2194.27783203125
Iteration 1333 logL: -2557.77099609375
Iteration 1334 logL: -2421.5693359375
```

```
Iteration 1335 logL: -1892.5177001953125
Iteration 1336 logL: -2038.3154296875
Iteration 1337 logL: -2469.393798828125
Iteration 1338 logL: -2282.8388671875
Iteration 1339 logL: -2071.3798828125
Iteration 1340 logL: -2266.095947265625
Iteration 1341 logL: -2383.323974609375
Iteration 1342 logL: -1980.119384765625
Iteration 1343 logL: -2199.6904296875
Iteration 1344 logL: -1928.39453125
Iteration 1345 logL: -2262.67578125
Iteration 1346 logL: -1975.611572265625
Iteration 1347 logL: -1961.2841796875
Iteration 1348 logL: -2014.4932861328125
Iteration 1349 logL: -2338.93408203125
Iteration 1350 logL: -2150.32421875
Iteration 1351 logL: -1841.7650146484375
Iteration 1352 logL: -1846.023681640625
Iteration 1353 logL: -1997.0467529296875
Iteration 1354 logL: -2061.569091796875
Iteration 1355 logL: -2134.61865234375
Iteration 1356 logL: -1864.1640625
Iteration 1357 logL: -2076.2958984375
Iteration 1358 logL: -1975.04345703125
Iteration 1359 logL: -2155.977783203125
Iteration 1360 logL: -2397.01513671875
Iteration 1361 logL: -2151.866943359375
Iteration 1362 logL: -2528.355712890625
Iteration 1363 logL: -1940.532958984375
Iteration 1364 logL: -1806.6131591796875
Iteration 1365 logL: -2124.21142578125
Iteration 1366 logL: -2439.08056640625
Iteration 1367 logL: -2179.723388671875
Iteration 1368 logL: -2853.701171875
Iteration 1369 logL: -1885.2803955078125
Iteration 1370 logL: -2280.45751953125
Iteration 1371 logL: -1804.5162353515625
Iteration 1372 logL: -2049.10009765625
Iteration 1373 logL: -1822.34130859375
Iteration 1374 logL: -2229.55322265625
Iteration 1375 logL: -2237.5185546875
Iteration 1376 logL: -1841.512939453125
Iteration 1377 logL: -1957.8353271484375
Iteration 1378 logL: -2063.21923828125
Iteration 1379 logL: -1903.84765625
Iteration 1380 logL: -1870.502197265625
Iteration 1381 logL: -1802.6282958984375
Iteration 1382 logL: -1928.4954833984375
Iteration 1383 logL: -2085.740478515625
Iteration 1384 logL: -2060.250244140625
Iteration 1385 logL: -1849.54833984375
Iteration 1386 logL: -1761.095458984375
Iteration 1387 logL: -2043.841064453125
Iteration 1388 logL: -1955.5052490234375
Iteration 1389 logL: -1913.1424560546875
Iteration 1390 logL: -1740.4129638671875
Iteration 1391 logL: -1918.1048583984375
Iteration 1392 logL: -1945.115234375
Iteration 1393 logL: -1826.45751953125
```

```
Iteration 1394 logL: -1926.200439453125
Iteration 1395 logL: -2111.3984375
Iteration 1396 logL: -1819.70751953125
Iteration 1397 logL: -2028.692138671875
Iteration 1398 logL: -1822.5328369140625
Iteration 1399 logL: -1863.4000244140625
Iteration 1400 logL: -1637.2442626953125
Iteration 1401 logL: -1799.9224853515625
Iteration 1402 logL: -1783.120361328125
Iteration 1403 logL: -1730.0948486328125
Iteration 1404 logL: -1722.510009765625
Iteration 1405 logL: -1840.315673828125
Iteration 1406 logL: -1861.749755859375
Iteration 1407 logL: -1974.2847900390625
Iteration 1408 logL: -1611.9000244140625
Iteration 1409 logL: -1592.41357421875
Iteration 1410 logL: -1659.760986328125
Iteration 1411 logL: -1708.6917724609375
Iteration 1412 logL: -1791.63330078125
Iteration 1413 logL: -1982.625
Iteration 1414 logL: -1746.8988037109375
Iteration 1415 logL: -1587.4288330078125
Iteration 1416 logL: -1707.8338623046875
Iteration 1417 logL: -1558.8341064453125
Iteration 1418 logL: -1807.6337890625
Iteration 1419 logL: -1855.7430419921875
Iteration 1420 logL: -1612.2431640625
Iteration 1421 logL: -1496.451904296875
Iteration 1422 logL: -1634.6871337890625
Iteration 1423 logL: -1699.23876953125
Iteration 1424 logL: -1433.310546875
Iteration 1425 logL: -2087.545654296875
Iteration 1426 logL: -1778.0235595703125
Iteration 1427 logL: -1630.5809326171875
Iteration 1428 logL: -1741.54931640625
Iteration 1429 logL: -2053.89501953125
Iteration 1430 logL: -1629.916015625
Iteration 1431 logL: -1899.0697021484375
Iteration 1432 logL: -1566.3096923828125
Iteration 1433 logL: -1769.9688720703125
Iteration 1434 logL: -1726.865966796875
Iteration 1435 logL: -1372.5416259765625
Iteration 1436 logL: -1572.54296875
Iteration 1437 logL: -1625.1290283203125
Iteration 1438 logL: -1616.625244140625
Iteration 1439 logL: -1503.3514404296875
Iteration 1440 logL: -1725.6246337890625
Iteration 1441 logL: -1632.8486328125
Iteration 1442 logL: -1770.5526123046875
Iteration 1443 logL: -1538.2828369140625
Iteration 1444 logL: -1591.120361328125
Iteration 1445 logL: -1755.9793701171875
Iteration 1446 logL: -1510.6171875
Iteration 1447 logL: -1461.1795654296875
Iteration 1448 logL: -1988.814453125
Iteration 1449 logL: -1436.876953125
Iteration 1450 logL: -2046.894287109375
Iteration 1451 logL: -1446.7579345703125
Iteration 1452 logL: -2000.81396484375
```

```
Iteration 1453 logL: -1787.935302734375
Iteration 1454 logL: -1621.5318603515625
Iteration 1455 logL: -1751.6385498046875
Iteration 1456 logL: -1657.4967041015625
Iteration 1457 logL: -2053.25341796875
Iteration 1458 logL: -2077.7216796875
Iteration 1459 logL: -1547.28125
Iteration 1460 logL: -1536.0279541015625
Iteration 1461 logL: -2059.982421875
Iteration 1462 logL: -1694.4859619140625
Iteration 1463 logL: -1587.8292236328125
Iteration 1464 logL: -1544.15234375
Iteration 1465 logL: -1474.68310546875
Iteration 1466 logL: -2005.8447265625
Iteration 1467 logL: -1611.0162353515625
Iteration 1468 logL: -1949.4326171875
Iteration 1469 logL: -1775.4090576171875
Iteration 1470 logL: -1642.955810546875
Iteration 1471 logL: -1543.5574951171875
Iteration 1472 logL: -1507.2550048828125
Iteration 1473 logL: -1962.774658203125
Iteration 1474 logL: -1951.442626953125
Iteration 1475 logL: -1543.9195556640625
Iteration 1476 logL: -1537.17919921875
Iteration 1477 logL: -2000.9818115234375
Iteration 1478 logL: -1628.2855224609375
Iteration 1479 logL: -2019.587158203125
Iteration 1480 logL: -1864.6142578125
Iteration 1481 logL: -1631.5311279296875
Iteration 1482 logL: -1745.202392578125
Iteration 1483 logL: -1681.5196533203125
Iteration 1484 logL: -1547.250732421875
Iteration 1485 logL: -1698.892333984375
Iteration 1486 logL: -1725.587890625
Iteration 1487 logL: -1513.69091796875
Iteration 1488 logL: -1722.861572265625
Iteration 1489 logL: -1812.584228515625
Iteration 1490 logL: -1699.245849609375
Iteration 1491 logL: -1800.2274169921875
Iteration 1492 logL: -1457.4163818359375
Iteration 1493 logL: -1580.4862060546875
Iteration 1494 logL: -1708.2337646484375
Iteration 1495 logL: -1851.9249267578125
Iteration 1496 logL: -1803.3653564453125
Iteration 1497 logL: -1695.4095458984375
Iteration 1498 logL: -1402.0606689453125
Iteration 1499 logL: -1353.5106201171875
Iteration 1500 logL: -1806.699462890625
Iteration 1501 logL: -1590.3121337890625
Iteration 1502 logL: -1612.9710693359375
Iteration 1503 logL: -1821.321044921875
Iteration 1504 logL: -1399.794677734375
Iteration 1505 logL: -1445.2486572265625
Iteration 1506 logL: -1864.868896484375
Iteration 1507 logL: -1477.126220703125
Iteration 1508 logL: -1371.7271728515625
Iteration 1509 logL: -1326.0369873046875
Iteration 1510 logL: -1645.328369140625
Iteration 1511 logL: -1518.280517578125
```

```
Iteration 1512 logL: -1586.7552490234375
Iteration 1513 logL: -1616.98291015625
Iteration 1514 logL: -1923.52783203125
Iteration 1515 logL: -1654.48681640625
Iteration 1516 logL: -1658.761474609375
Iteration 1517 logL: -1450.9932861328125
Iteration 1518 logL: -1853.2845458984375
Iteration 1519 logL: -1485.3402099609375
Iteration 1520 logL: -1551.3311767578125
Iteration 1521 logL: -1443.622802734375
Iteration 1522 logL: -1636.74853515625
Iteration 1523 logL: -1538.3182373046875
Iteration 1524 logL: -1416.1851806640625
Iteration 1525 logL: -1480.7850341796875
Iteration 1526 logL: -1566.7340087890625
Iteration 1527 logL: -1645.4052734375
Iteration 1528 logL: -1602.4827880859375
Iteration 1529 logL: -1453.347412109375
Iteration 1530 logL: -1614.96044921875
Iteration 1531 logL: -1401.040771484375
Iteration 1532 logL: -1524.4293212890625
Iteration 1533 logL: -1426.799072265625
Iteration 1534 logL: -1697.10400390625
Iteration 1535 logL: -1547.6968994140625
Iteration 1536 logL: -1431.9581298828125
Iteration 1537 logL: -1292.1644287109375
Iteration 1538 logL: -1685.7489013671875
Iteration 1539 logL: -1501.3426513671875
Iteration 1540 logL: -1464.48681640625
Iteration 1541 logL: -1309.6083984375
Iteration 1542 logL: -1425.254638671875
Iteration 1543 logL: -1417.8509521484375
Iteration 1544 logL: -1684.3265380859375
Iteration 1545 logL: -1566.008056640625
Iteration 1546 logL: -1320.541259765625
Iteration 1547 logL: -1593.4884033203125
Iteration 1548 logL: -1413.1441650390625
Iteration 1549 logL: -1666.5958251953125
Iteration 1550 logL: -1596.881103515625
Iteration 1551 logL: -2033.43212890625
Iteration 1552 logL: -1487.9202880859375
Iteration 1553 logL: -1258.6005859375
Iteration 1554 logL: -1513.0421142578125
Iteration 1555 logL: -1532.954345703125
Iteration 1556 logL: -1267.11962890625
Iteration 1557 logL: -1336.03125
Iteration 1558 logL: -1526.9178466796875
Iteration 1559 logL: -1216.7047119140625
Iteration 1560 logL: -1521.1041259765625
Iteration 1561 logL: -1942.0791015625
Iteration 1562 logL: -1336.9322509765625
Iteration 1563 logL: -1308.4619140625
Iteration 1564 logL: -1560.1959228515625
Iteration 1565 logL: -1554.3807373046875
Iteration 1566 logL: -1589.8492431640625
Iteration 1567 logL: -1712.0570068359375
Iteration 1568 logL: -1706.903076171875
Iteration 1569 logL: -1190.98583984375
Iteration 1570 logL: -1395.9580078125
```

```
Iteration 1571 logL: -1625.474365234375
Iteration 1572 logL: -1618.5718994140625
Iteration 1573 logL: -1234.91064453125
Iteration 1574 logL: -1581.50244140625
Iteration 1575 logL: -1373.535888671875
Iteration 1576 logL: -1417.5321044921875
Iteration 1577 logL: -1356.541015625
Iteration 1578 logL: -1387.89794921875
Iteration 1579 logL: -1476.1348876953125
Iteration 1580 logL: -1611.224853515625
Iteration 1581 logL: -1585.50390625
Iteration 1582 logL: -1327.4388427734375
Iteration 1583 logL: -1844.7926025390625
Iteration 1584 logL: -1261.28662109375
Iteration 1585 logL: -1229.953125
Iteration 1586 logL: -1522.195068359375
Iteration 1587 logL: -1382.7174072265625
Iteration 1588 logL: -1339.97314453125
Iteration 1589 logL: -1527.83251953125
Iteration 1590 logL: -1491.3065185546875
Iteration 1591 logL: -1320.4254150390625
Iteration 1592 logL: -1310.841064453125
Iteration 1593 logL: -1368.5462646484375
Iteration 1594 logL: -1680.034912109375
Iteration 1595 logL: -1416.6622314453125
Iteration 1596 logL: -1478.2900390625
Iteration 1597 logL: -1340.971923828125
Iteration 1598 logL: -1530.1341552734375
Iteration 1599 logL: -1526.07421875
Iteration 1600 logL: -1351.04736328125
Iteration 1601 logL: -1375.83837890625
Iteration 1602 logL: -1287.29931640625
Iteration 1603 logL: -1401.554931640625
Iteration 1604 logL: -1353.2379150390625
Iteration 1605 logL: -1362.84326171875
Iteration 1606 logL: -1152.98193359375
Iteration 1607 logL: -1396.9176025390625
Iteration 1608 logL: -1176.5584716796875
Iteration 1609 logL: -1308.38427734375
Iteration 1610 logL: -1233.887939453125
Iteration 1611 logL: -1541.0308837890625
Iteration 1612 logL: -1263.168701171875
Iteration 1613 logL: -1354.600830078125
Iteration 1614 logL: -1382.578857421875
Iteration 1615 logL: -1137.2315673828125
Iteration 1616 logL: -1158.908935546875
Iteration 1617 logL: -1446.739990234375
Iteration 1618 logL: -1305.9072265625
Iteration 1619 logL: -1488.6180419921875
Iteration 1620 logL: -1392.830810546875
Iteration 1621 logL: -1195.819091796875
Iteration 1622 logL: -1361.3712158203125
Iteration 1623 logL: -1226.637451171875
Iteration 1624 logL: -1712.0093994140625
Iteration 1625 logL: -1148.0943603515625
Iteration 1626 logL: -1270.4459228515625
Iteration 1627 logL: -1668.4578857421875
Iteration 1628 logL: -1570.2254638671875
Iteration 1629 logL: -1443.7271728515625
```

```
Iteration 1630 logL: -1108.245361328125
Iteration 1631 logL: -980.154541015625
Iteration 1632 logL: -1698.7410888671875
Iteration 1633 logL: -1757.3497314453125
Iteration 1634 logL: -1512.923828125
Iteration 1635 logL: -1454.6019287109375
Iteration 1636 logL: -1216.51708984375
Iteration 1637 logL: -1537.007080078125
Iteration 1638 logL: -1231.947509765625
Iteration 1639 logL: -1280.5615234375
Iteration 1640 logL: -1376.691650390625
Iteration 1641 logL: -1262.2015380859375
Iteration 1642 logL: -1357.578857421875
Iteration 1643 logL: -1283.6619873046875
Iteration 1644 logL: -1173.1424560546875
Iteration 1645 logL: -1601.4505615234375
Iteration 1646 logL: -1226.6002197265625
Iteration 1647 logL: -1075.4423828125
Iteration 1648 logL: -1079.093017578125
Iteration 1649 logL: -1268.52294921875
Iteration 1650 logL: -1251.756103515625
Iteration 1651 logL: -1119.5074462890625
Iteration 1652 logL: -1142.4112548828125
Iteration 1653 logL: -1289.8638916015625
Iteration 1654 logL: -1333.725341796875
Iteration 1655 logL: -1064.036376953125
Iteration 1656 logL: -1090.552734375
Iteration 1657 logL: -1201.73193359375
Iteration 1658 logL: -1215.242431640625
Iteration 1659 logL: -1205.101806640625
Iteration 1660 logL: -1190.859619140625
Iteration 1661 logL: -1583.8828125
Iteration 1662 logL: -1381.877197265625
Iteration 1663 logL: -1119.6689453125
Iteration 1664 logL: -1395.594970703125
Iteration 1665 logL: -1469.769287109375
Iteration 1666 logL: -1297.427490234375
Iteration 1667 logL: -1843.0980224609375
Iteration 1668 logL: -1170.1298828125
Iteration 1669 logL: -1132.216064453125
Iteration 1670 logL: -1731.2611083984375
Iteration 1671 logL: -1832.781005859375
Iteration 1672 logL: -1544.6690673828125
Iteration 1673 logL: -1364.1805419921875
Iteration 1674 logL: -1295.0057373046875
Iteration 1675 logL: -1440.8431396484375
Iteration 1676 logL: -1243.4681396484375
Iteration 1677 logL: -1177.9171142578125
Iteration 1678 logL: -1447.6947021484375
Iteration 1679 logL: -1323.152099609375
Iteration 1680 logL: -1220.2193603515625
Iteration 1681 logL: -1355.2041015625
Iteration 1682 logL: -1195.4498291015625
Iteration 1683 logL: -1440.907958984375
Iteration 1684 logL: -985.4444580078125
Iteration 1685 logL: -1490.2255859375
Iteration 1686 logL: -1177.082275390625
Iteration 1687 logL: -1277.21875
Iteration 1688 logL: -1075.8192138671875
```

```
Iteration 1689 logL: -1519.3946533203125
Iteration 1690 logL: -1464.6429443359375
Iteration 1691 logL: -1195.5570068359375
Iteration 1692 logL: -1216.995849609375
Iteration 1693 logL: -2036.31787109375
Iteration 1694 logL: -1192.2882080078125
Iteration 1695 logL: -1466.5367431640625
Iteration 1696 logL: -1394.9813232421875
Iteration 1697 logL: -1106.302978515625
Iteration 1698 logL: -1058.5843505859375
Iteration 1699 logL: -1213.614501953125
Iteration 1700 logL: -1297.6826171875
Iteration 1701 logL: -1075.9617919921875
Iteration 1702 logL: -1110.753662109375
Iteration 1703 logL: -1012.0301513671875
Iteration 1704 logL: -1062.61328125
Iteration 1705 logL: -972.349365234375
Iteration 1706 logL: -1149.63916015625
Iteration 1707 logL: -1136.0001220703125
Iteration 1708 logL: -1239.5062255859375
Iteration 1709 logL: -1314.9522705078125
Iteration 1710 logL: -1370.3277587890625
Iteration 1711 logL: -942.014404296875
Iteration 1712 logL: -1538.1478271484375
Iteration 1713 logL: -1420.830810546875
Iteration 1714 logL: -1113.2503662109375
Iteration 1715 logL: -1071.1942138671875
Iteration 1716 logL: -1133.641845703125
Iteration 1717 logL: -1045.2742919921875
Iteration 1718 logL: -1122.6177978515625
Iteration 1719 logL: -983.59765625
Iteration 1720 logL: -1099.0838623046875
Iteration 1721 logL: -1309.4378662109375
Iteration 1722 logL: -1249.0245361328125
Iteration 1723 logL: -1070.9197998046875
Iteration 1724 logL: -1232.697021484375
Iteration 1725 logL: -1232.0894775390625
Iteration 1726 logL: -1573.03173828125
Iteration 1727 logL: -1285.503662109375
Iteration 1728 logL: -1222.336669921875
Iteration 1729 logL: -1338.29736328125
Iteration 1730 logL: -1187.7841796875
Iteration 1731 logL: -1015.5546875
Iteration 1732 logL: -1132.734375
Iteration 1733 logL: -1316.859375
Iteration 1734 logL: -1063.3460693359375
Iteration 1735 logL: -1030.776123046875
Iteration 1736 logL: -1258.1480712890625
Iteration 1737 logL: -1177.28662109375
Iteration 1738 logL: -920.7503662109375
Iteration 1739 logL: -1179.2530517578125
Iteration 1740 logL: -1270.06396484375
Iteration 1741 logL: -1358.6314697265625
Iteration 1742 logL: -1324.02099609375
Iteration 1743 logL: -1041.6455078125
Iteration 1744 logL: -1105.0618896484375
Iteration 1745 logL: -1108.971435546875
Iteration 1746 logL: -1100.663818359375
Iteration 1747 logL: -1301.64111328125
```

```
Iteration 1748 logL: -1329.9981689453125
Iteration 1749 logL: -1207.31640625
Iteration 1750 logL: -1187.7232666015625
Iteration 1751 logL: -1354.99951171875
Iteration 1752 logL: -1052.1552734375
Iteration 1753 logL: -1007.591064453125
Iteration 1754 logL: -1387.25537109375
Iteration 1755 logL: -1172.95361328125
Iteration 1756 logL: -1114.8958740234375
Iteration 1757 logL: -1268.288330078125
Iteration 1758 logL: -1042.7626953125
Iteration 1759 logL: -1295.860595703125
Iteration 1760 logL: -1280.3271484375
Iteration 1761 logL: -1251.084228515625
Iteration 1762 logL: -1205.45361328125
Iteration 1763 logL: -1690.6689453125
Iteration 1764 logL: -1081.05322265625
Iteration 1765 logL: -1246.945068359375
Iteration 1766 logL: -1039.462890625
Iteration 1767 logL: -1142.111572265625
Iteration 1768 logL: -1265.45654296875
Iteration 1769 logL: -1281.721923828125
Iteration 1770 logL: -1309.653564453125
Iteration 1771 logL: -1680.562255859375
Iteration 1772 logL: -1293.718994140625
Iteration 1773 logL: -865.4189453125
Iteration 1774 logL: -1144.5948486328125
Iteration 1775 logL: -1170.566162109375
Iteration 1776 logL: -1996.01025390625
Iteration 1777 logL: -1267.449462890625
Iteration 1778 logL: -1196.176025390625
Iteration 1779 logL: -1323.94482421875
Iteration 1780 logL: -1565.8349609375
Iteration 1781 logL: -1299.803955078125
Iteration 1782 logL: -1372.863525390625
Iteration 1783 logL: -1746.3642578125
Iteration 1784 logL: -1097.82568359375
Iteration 1785 logL: -1432.067138671875
Iteration 1786 logL: -1337.90478515625
Iteration 1787 logL: -1412.645751953125
Iteration 1788 logL: -1321.851806640625
Iteration 1789 logL: -1345.647216796875
Iteration 1790 logL: -1445.92724609375
Iteration 1791 logL: -1402.366943359375
Iteration 1792 logL: -1607.05810546875
Iteration 1793 logL: -1393.985595703125
Iteration 1794 logL: -1226.673828125
Iteration 1795 logL: -1044.9970703125
Iteration 1796 logL: -1135.041748046875
Iteration 1797 logL: -1444.684814453125
Iteration 1798 logL: -1373.3154296875
Iteration 1799 logL: -1316.138427734375
Iteration 1800 logL: -1214.371826171875
Iteration 1801 logL: -1363.731689453125
Iteration 1802 logL: -1120.855224609375
Iteration 1803 logL: -1385.364501953125
Iteration 1804 logL: -1125.908447265625
Iteration 1805 logL: -1044.094482421875
Iteration 1806 logL: -1195.940673828125
```

```
Iteration 1807 logL: -1424.763671875
Iteration 1808 logL: -1256.38623046875
Iteration 1809 logL: -931.464599609375
Iteration 1810 logL: -1171.986083984375
Iteration 1811 logL: -1136.90380859375
Iteration 1812 logL: -1266.07421875
Iteration 1813 logL: -1316.348876953125
Iteration 1814 logL: -1142.283935546875
Iteration 1815 logL: -1147.531982421875
Iteration 1816 logL: -1013.98046875
Iteration 1817 logL: -1012.45703125
Iteration 1818 logL: -1040.140380859375
Iteration 1819 logL: -1144.924560546875
Iteration 1820 logL: -991.103515625
Iteration 1821 logL: -1134.0400390625
Iteration 1822 logL: -1118.8193359375
Iteration 1823 logL: -1024.966064453125
Iteration 1824 logL: -1317.71630859375
Iteration 1825 logL: -1428.437255859375
Iteration 1826 logL: -1159.018798828125
Iteration 1827 logL: -909.503173828125
Iteration 1828 logL: -1202.47509765625
Iteration 1829 logL: -1545.25634765625
Iteration 1830 logL: -1346.7109375
Iteration 1831 logL: -919.37353515625
Iteration 1832 logL: -1143.79931640625
Iteration 1833 logL: -1081.80712890625
Iteration 1834 logL: -1079.048828125
Iteration 1835 logL: -1599.61181640625
Iteration 1836 logL: -1262.09765625
Iteration 1837 logL: -1176.054931640625
Iteration 1838 logL: -1337.212890625
Iteration 1839 logL: -891.056396484375
Iteration 1840 logL: -1211.4208984375
Iteration 1841 logL: -1020.72265625
Iteration 1842 logL: -1543.94482421875
Iteration 1843 logL: -1057.576416015625
Iteration 1844 logL: -898.874755859375
Iteration 1845 logL: -1209.9482421875
Iteration 1846 logL: -1551.938720703125
Iteration 1847 logL: -1115.676513671875
Iteration 1848 logL: -1101.764892578125
Iteration 1849 logL: -1119.8642578125
Iteration 1850 logL: -1010.772216796875
Iteration 1851 logL: -1057.302734375
Iteration 1852 logL: -1090.306884765625
Iteration 1853 logL: -1407.312255859375
Iteration 1854 logL: -1477.843505859375
Iteration 1855 logL: -1237.176025390625
Iteration 1856 logL: -1005.63037109375
Iteration 1857 logL: -1027.980224609375
Iteration 1858 logL: -1343.48583984375
Iteration 1859 logL: -1054.12548828125
Iteration 1860 logL: -1100.2314453125
Iteration 1861 logL: -940.71484375
Iteration 1862 logL: -1200.455322265625
Iteration 1863 logL: -1142.964111328125
Iteration 1864 logL: -905.65185546875
Iteration 1865 logL: -1021.826416015625
```

```
Iteration 1866 logL: -1014.808349609375
Iteration 1867 logL: -1071.56103515625
Iteration 1868 logL: -1333.482177734375
Iteration 1869 logL: -963.418701171875
Iteration 1870 logL: -1064.89697265625
Iteration 1871 logL: -1108.72900390625
Iteration 1872 logL: -1067.771484375
Iteration 1873 logL: -1127.8828125
Iteration 1874 logL: -1081.60302734375
Iteration 1875 logL: -965.2626953125
Iteration 1876 logL: -988.87255859375
Iteration 1877 logL: -1065.111083984375
Iteration 1878 logL: -998.36376953125
Iteration 1879 logL: -987.1142578125
Iteration 1880 logL: -1086.422119140625
Iteration 1881 logL: -1011.31591796875
Iteration 1882 logL: -1096.4521484375
Iteration 1883 logL: -1229.218994140625
Iteration 1884 logL: -1258.369873046875
Iteration 1885 logL: -1353.3291015625
Iteration 1886 logL: -857.21875
Iteration 1887 logL: -1257.5390625
Iteration 1888 logL: -983.54736328125
Iteration 1889 logL: -1463.050048828125
Iteration 1890 logL: -902.812255859375
Iteration 1891 logL: -1015.60205078125
Iteration 1892 logL: -937.58642578125
Iteration 1893 logL: -973.899169921875
Iteration 1894 logL: -1120.499755859375
Iteration 1895 logL: -1576.31640625
Iteration 1896 logL: -1102.259521484375
Iteration 1897 logL: -1324.40478515625
Iteration 1898 logL: -1041.040771484375
Iteration 1899 logL: -1416.749267578125
Iteration 1900 logL: -1493.891357421875
Iteration 1901 logL: -1039.11279296875
Iteration 1902 logL: -1012.100341796875
Iteration 1903 logL: -1212.315673828125
Iteration 1904 logL: -1539.478515625
Iteration 1905 logL: -1077.4384765625
Iteration 1906 logL: -992.618408203125
Iteration 1907 logL: -979.22802734375
Iteration 1908 logL: -1281.15771484375
Iteration 1909 logL: -1554.8798828125
Iteration 1910 logL: -1005.364990234375
Iteration 1911 logL: -899.75537109375
Iteration 1912 logL: -1055.173095703125
Iteration 1913 logL: -1163.792724609375
Iteration 1914 logL: -1157.046142578125
Iteration 1915 logL: -1776.942626953125
Iteration 1916 logL: -1032.344970703125
Iteration 1917 logL: -1102.173095703125
Iteration 1918 logL: -1496.106201171875
Iteration 1919 logL: -1098.893310546875
Iteration 1920 logL: -1040.317138671875
Iteration 1921 logL: -1673.8388671875
Iteration 1922 logL: -1684.88623046875
Iteration 1923 logL: -1116.3251953125
Iteration 1924 logL: -1416.927001953125
```

```
Iteration 1925 logL: -987.152587890625
Iteration 1926 logL: -1066.673095703125
Iteration 1927 logL: -1040.917724609375
Iteration 1928 logL: -1074.169921875
Iteration 1929 logL: -1119.952880859375
Iteration 1930 logL: -1414.101806640625
Iteration 1931 logL: -876.994384765625
Iteration 1932 logL: -971.498046875
Iteration 1933 logL: -1136.347412109375
Iteration 1934 logL: -1456.970703125
Iteration 1935 logL: -1219.20166015625
Iteration 1936 logL: -991.05908203125
Iteration 1937 logL: -1131.446044921875
Iteration 1938 logL: -944.871337890625
Iteration 1939 logL: -915.4677734375
Iteration 1940 logL: -1117.62548828125
Iteration 1941 logL: -1233.37841796875
Iteration 1942 logL: -1017.934814453125
Iteration 1943 logL: -832.17529296875
Iteration 1944 logL: -989.371826171875
Iteration 1945 logL: -1345.60302734375
Iteration 1946 logL: -1143.80224609375
Iteration 1947 logL: -1175.467041015625
Iteration 1948 logL: -1102.8701171875
Iteration 1949 logL: -1056.500244140625
Iteration 1950 logL: -951.403564453125
Iteration 1951 logL: -856.87353515625
Iteration 1952 logL: -1007.120849609375
Iteration 1953 logL: -896.989013671875
Iteration 1954 logL: -1194.041259765625
Iteration 1955 logL: -1053.199462890625
Iteration 1956 logL: -865.03466796875
Iteration 1957 logL: -1246.1025390625
Iteration 1958 logL: -1341.726318359375
Iteration 1959 logL: -889.03857421875
Iteration 1960 logL: -1292.580322265625
Iteration 1961 logL: -921.563232421875
Iteration 1962 logL: -1567.103759765625
Iteration 1963 logL: -1208.059814453125
Iteration 1964 logL: -921.08056640625
Iteration 1965 logL: -1033.3154296875
Iteration 1966 logL: -1243.2275390625
Iteration 1967 logL: -1147.265380859375
Iteration 1968 logL: -1195.745361328125
Iteration 1969 logL: -827.04443359375
Iteration 1970 logL: -1138.609375
Iteration 1971 logL: -975.8818359375
Iteration 1972 logL: -1208.3505859375
Iteration 1973 logL: -855.1357421875
Iteration 1974 logL: -957.8818359375
Iteration 1975 logL: -963.307373046875
Iteration 1976 logL: -1086.295166015625
Iteration 1977 logL: -902.845703125
Iteration 1978 logL: -1262.313232421875
Iteration 1979 logL: -1039.027587890625
Iteration 1980 logL: -892.798095703125
Iteration 1981 logL: -943.1220703125
Iteration 1982 logL: -1156.06591796875
Iteration 1983 logL: -1353.580078125
```

```
Iteration 1984 logL: -856.92529296875
Iteration 1985 logL: -1020.078857421875
Iteration 1986 logL: -1095.52685546875
Iteration 1987 logL: -1237.697265625
Iteration 1988 logL: -998.10693359375
Iteration 1989 logL: -1152.889892578125
Iteration 1990 logL: -963.375732421875
Iteration 1991 logL: -925.7431640625
Iteration 1992 logL: -906.114013671875
Iteration 1993 logL: -1085.943359375
Iteration 1994 logL: -1247.585693359375
Iteration 1995 logL: -931.400146484375
Iteration 1996 logL: -993.2607421875
Iteration 1997 logL: -1423.531005859375
Iteration 1998 logL: -968.134521484375
Iteration 1999 logL: -927.2412109375
Iteration 2000 logL: -987.966064453125
```

### Use prediction to visualize the resulting BNN

```
In [15]: xt = np.linspace(0,1,100)[:,None]
```

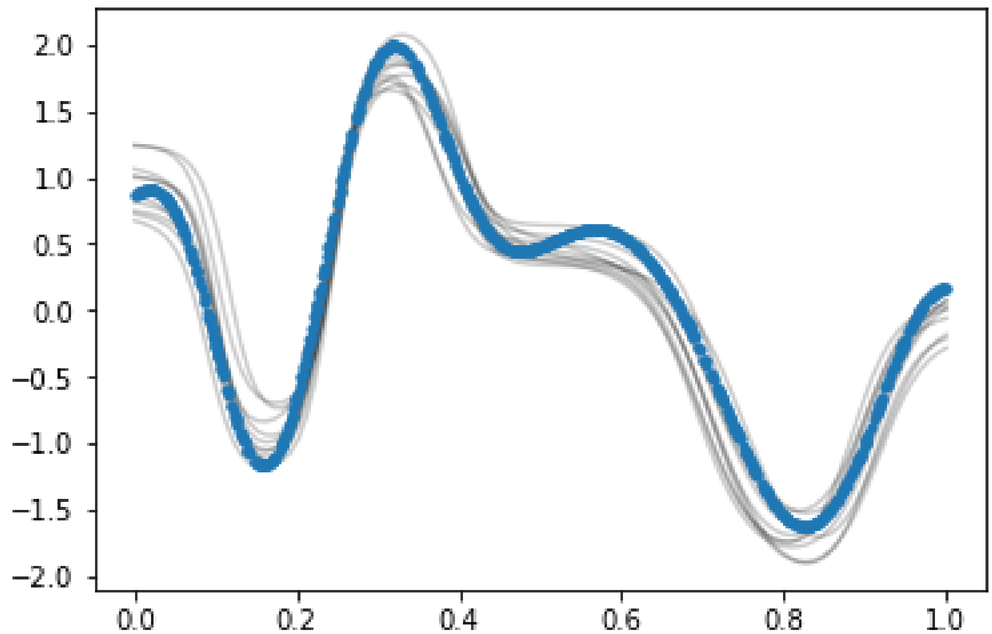
```
In [16]: infr2 = VariationalPosteriorForwardSampling(10, [m.x], infr, [m.r])
         res = infr2.run(x=mx.nd.array(xt))
```

```
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/core/factor_graph.py:65: UserWarning:The value N has
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/core/factor_graph.py:65: UserWarning:The value y has
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/inference/inference_parameters.py:52: UserWarning:In
```

```
In [17]: yt = res[m.r].asnumpy()
```

```
In [18]: # plot(xt[:,0],yt[:,0])
         yt_mean = yt.mean(0)
         yt_std = yt.std(0)
         #plot(xt[:,0], yt.mean(0)[:,0])
         #errorbar(xt[:,0],y=yt_mean[:,0],yerr=yt_std[:,0]*2)
         for i in range(yt.shape[0]):
             plot(xt[:,0],yt[i,:,0],'k',alpha=0.2)
         plot(x[:,0],y[:,0],'.')
```

```
Out[18]: [<matplotlib.lines.Line2D at 0x1a26059b00>]
```




### 4.2.3 Variational Auto-Encoder (VAE)

#### Zhenwen Dai (2018-8-21)

```
In [1]: import mxfusion as mf
        import mxnet as mx
        import numpy as np
        import mxnet.gluon.nn as nn
        import mxfusion.components
        import mxfusion.inference
        %matplotlib inline
        from pylab import *
```

```
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion
  from ._conv import register_converters as _register_converters
```

#### Load a toy dataset

```
In [2]: import GPy
        data = GPy.util.datasets.oil_100()
        Y = data['X']
        label = data['Y'].argmax(1)
```

```
In [3]: N, D = Y.shape
```

#### Model Defintion

```
In [4]: Q = 2
```

```
In [5]: H = 50
        encoder = nn.HybridSequential(prefix='encoder_')
        with encoder.name_scope():
```

```
        encoder.add(nn.Dense(H, activation="tanh"))
        encoder.add(nn.Dense(H, activation="tanh"))
        encoder.add(nn.Dense(Q, flatten=True))
    encoder.initialize(mx.init.Xavier(magnitude=3))
    _=encoder(mx.nd.array(np.random.rand(5,D)))
```

```
In [6]: H = 50
        decoder = nn.HybridSequential(prefix='decoder_')
        with decoder.name_scope():
            decoder.add(nn.Dense(H, activation="tanh"))
            decoder.add(nn.Dense(H, activation="tanh"))
            decoder.add(nn.Dense(D, flatten=True))
        decoder.initialize(mx.init.Xavier(magnitude=3))
        _=decoder(mx.nd.array(np.random.rand(5,Q)))
```

```
In [7]: from mxfusion.components.var_trans import PositiveTransformation
```

```
In [8]: m = mf.models.Model()
        m.N = mf.components.Variable()
        m.decoder = mf.components.functions.MXFusionGluonFunction(decoder, nOutputs=1,broadcastable=F
        m.x = mf.components.distributions.Normal.define_variable(mean=mx.nd.array([0]), variance=mx.n
        m.f = m.decoder(m.x)
        m.noise_var = mf.components.Variable(shape=(1,), transformation=PositiveTransformation(), ini
        m.y = mf.components.distributions.Normal.define_variable(mean=m.f, variance=m.noise_var, shap
        m.show()
```

```
x ~ Normal(mean=Variable(61c66), variance=Variable(3581c))
f = GluonFunctionEvaluation(decoder_dense0_weight=Variable(1e112), decoder_dense0_bias=Variable(12aa9
y ~ Normal(mean=f, variance=noise_var)
```

```
In [9]: q = mf.models.Posterior(m)
        q.x_var = mf.components.Variable(shape=(1,), transformation=PositiveTransformation(), initial
        q.encoder = mf.components.functions.MXFusionGluonFunction(encoder, nOutputs=1, broadcastable=
        q.x_mean = q.encoder(q.y)
        q.x.set_prior(mf.components.distributions.Normal(mean=q.x_mean, variance=q.x_var))
        q.show()
```

```
x_mean = GluonFunctionEvaluation(encoder_dense0_weight=Variable(2b695), encoder_dense0_bias=Variable
x ~ Normal(mean=x_mean, variance=x_var)
```

### Variational Inference

```
In [10]: from mxfusion.inference import BatchInferenceLoop, StochasticVariationalInference, GradBased
```

```
In [11]: observed = [m.y]
         alg = StochasticVariationalInference(num_samples=3, model=m, posterior=q, observed=observed)
         infr = GradBasedInference(inference_algorithm=alg, grad_loop=BatchInferenceLoop())
```

```
In [12]: infr.run(max_iter=2000, learning_rate=1e-2, y=mx.nd.array(Y), verbose=True)
```

```
/Users/zhenwend/mxfusion/src/MXFusion/mxfusion/inference/inference_parameters.py:52: UserWarning:In
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
```

```
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
/Users/zhenwend/anaconda3/lib/python3.6/site-packages/mxnet/gluon/parameter.py:689: UserWarning:Para
```

```
Iteration 1 logL: -1460.925048828125
Iteration 2 logL: -1562.58935546875
Iteration 3 logL: -1281.037109375
Iteration 4 logL: -1202.09814453125
Iteration 5 logL: -1215.446533203125
Iteration 6 logL: -1242.7581787109375
Iteration 7 logL: -1206.203125
Iteration 8 logL: -1193.624755859375
Iteration 9 logL: -1133.6195068359375
Iteration 10 logL: -1112.78466796875
Iteration 11 logL: -1098.210205078125
Iteration 12 logL: -1090.024169921875
Iteration 13 logL: -1077.6949462890625
Iteration 14 logL: -1074.197021484375
Iteration 15 logL: -1060.966796875
Iteration 16 logL: -1047.22314453125
Iteration 17 logL: -1042.4344482421875
Iteration 18 logL: -1035.609619140625
Iteration 19 logL: -1028.633544921875
Iteration 20 logL: -1029.137939453125
Iteration 21 logL: -1024.3011474609375
Iteration 22 logL: -1023.6302490234375
Iteration 23 logL: -1019.930908203125
Iteration 24 logL: -1010.139892578125
Iteration 25 logL: -1006.3385009765625
Iteration 26 logL: -1000.4537353515625
Iteration 27 logL: -994.9822998046875
Iteration 28 logL: -989.8155517578125
Iteration 29 logL: -982.1632080078125
Iteration 30 logL: -977.3480834960938
Iteration 31 logL: -974.651123046875
Iteration 32 logL: -974.5823364257812
Iteration 33 logL: -967.2603759765625
Iteration 34 logL: -963.084716796875
Iteration 35 logL: -957.8111572265625
Iteration 36 logL: -956.9771118164062
Iteration 37 logL: -947.5377807617188
Iteration 38 logL: -945.8042602539062
Iteration 39 logL: -944.1073608398438
Iteration 40 logL: -938.4964599609375
Iteration 41 logL: -930.6653442382812
Iteration 42 logL: -925.6103515625
Iteration 43 logL: -924.9813842773438
Iteration 44 logL: -924.310302734375
Iteration 45 logL: -916.4560546875
Iteration 46 logL: -917.2510375976562
Iteration 47 logL: -910.7885131835938
Iteration 48 logL: -908.1454467773438
Iteration 49 logL: -902.584716796875
Iteration 50 logL: -900.215087890625
Iteration 51 logL: -894.9979248046875
Iteration 52 logL: -892.5367431640625
Iteration 53 logL: -890.0333251953125
Iteration 54 logL: -885.6333618164062
Iteration 55 logL: -880.012451171875
Iteration 56 logL: -876.5682373046875
```

```
Iteration 57 logL: -872.11328125
Iteration 58 logL: -874.2131958007812
Iteration 59 logL: -866.4677734375
Iteration 60 logL: -862.5056762695312
Iteration 61 logL: -860.6942138671875
Iteration 62 logL: -859.4818115234375
Iteration 63 logL: -851.3370971679688
Iteration 64 logL: -846.3784790039062
Iteration 65 logL: -844.3232421875
Iteration 66 logL: -841.620361328125
Iteration 67 logL: -839.131591796875
Iteration 68 logL: -835.3382568359375
Iteration 69 logL: -831.16259765625
Iteration 70 logL: -828.3026123046875
Iteration 71 logL: -823.78564453125
Iteration 72 logL: -819.6796875
Iteration 73 logL: -816.356201171875
Iteration 74 logL: -812.4974365234375
Iteration 75 logL: -807.6212768554688
Iteration 76 logL: -804.9386596679688
Iteration 77 logL: -804.6951293945312
Iteration 78 logL: -796.6724243164062
Iteration 79 logL: -793.4720458984375
Iteration 80 logL: -790.3905029296875
Iteration 81 logL: -787.1576538085938
Iteration 82 logL: -786.882080078125
Iteration 83 logL: -786.7227172851562
Iteration 84 logL: -779.7852783203125
Iteration 85 logL: -774.4854125976562
Iteration 86 logL: -774.186767578125
Iteration 87 logL: -774.4632568359375
Iteration 88 logL: -761.4959716796875
Iteration 89 logL: -761.1930541992188
Iteration 90 logL: -757.3854370117188
Iteration 91 logL: -750.7093505859375
Iteration 92 logL: -749.3633422851562
Iteration 93 logL: -742.4610595703125
Iteration 94 logL: -745.5780639648438
Iteration 95 logL: -739.9026489257812
Iteration 96 logL: -735.1586303710938
Iteration 97 logL: -729.9743041992188
Iteration 98 logL: -727.3143920898438
Iteration 99 logL: -721.768798828125
Iteration 100 logL: -721.2698974609375
Iteration 101 logL: -714.2559204101562
Iteration 102 logL: -707.76318359375
Iteration 103 logL: -709.2598876953125
Iteration 104 logL: -706.3348999023438
Iteration 105 logL: -707.056884765625
Iteration 106 logL: -698.32958984375
Iteration 107 logL: -692.884765625
Iteration 108 logL: -688.1893310546875
Iteration 109 logL: -683.2183837890625
Iteration 110 logL: -684.6583251953125
Iteration 111 logL: -674.1200561523438
Iteration 112 logL: -674.0984497070312
Iteration 113 logL: -671.21533203125
Iteration 114 logL: -664.57421875
Iteration 115 logL: -666.19140625
```

```
Iteration 116 logL: -666.9966430664062
Iteration 117 logL: -651.857177734375
Iteration 118 logL: -658.775146484375
Iteration 119 logL: -650.1072998046875
Iteration 120 logL: -645.4615478515625
Iteration 121 logL: -648.34326171875
Iteration 122 logL: -640.14599609375
Iteration 123 logL: -636.5962524414062
Iteration 124 logL: -639.0338134765625
Iteration 125 logL: -626.3507080078125
Iteration 126 logL: -624.8009643554688
Iteration 127 logL: -618.8037109375
Iteration 128 logL: -618.6328125
Iteration 129 logL: -613.326171875
Iteration 130 logL: -603.6983032226562
Iteration 131 logL: -619.691162109375
Iteration 132 logL: -601.8302612304688
Iteration 133 logL: -601.2283325195312
Iteration 134 logL: -597.1408081054688
Iteration 135 logL: -592.0599975585938
Iteration 136 logL: -594.4515380859375
Iteration 137 logL: -588.57861328125
Iteration 138 logL: -585.05078125
Iteration 139 logL: -578.5697631835938
Iteration 140 logL: -578.210205078125
Iteration 141 logL: -575.7313232421875
Iteration 142 logL: -573.1090087890625
Iteration 143 logL: -569.6796875
Iteration 144 logL: -568.4530029296875
Iteration 145 logL: -567.4485473632812
Iteration 146 logL: -549.36376953125
Iteration 147 logL: -547.8933715820312
Iteration 148 logL: -553.80126953125
Iteration 149 logL: -548.371826171875
Iteration 150 logL: -546.1668090820312
Iteration 151 logL: -536.4398193359375
Iteration 152 logL: -532.5432739257812
Iteration 153 logL: -532.8011474609375
Iteration 154 logL: -531.6589965820312
Iteration 155 logL: -528.7539672851562
Iteration 156 logL: -524.3281860351562
Iteration 157 logL: -518.0106201171875
Iteration 158 logL: -516.4388427734375
Iteration 159 logL: -516.5404052734375
Iteration 160 logL: -513.127197265625
Iteration 161 logL: -501.44317626953125
Iteration 162 logL: -500.8021240234375
Iteration 163 logL: -508.0379638671875
Iteration 164 logL: -497.26849365234375
Iteration 165 logL: -499.436279296875
Iteration 166 logL: -493.8616943359375
Iteration 167 logL: -488.71337890625
Iteration 168 logL: -495.5947265625
Iteration 169 logL: -486.35211181640625
Iteration 170 logL: -483.2886962890625
Iteration 171 logL: -480.36505126953125
Iteration 172 logL: -482.5163269042969
Iteration 173 logL: -472.4610595703125
Iteration 174 logL: -484.8624267578125
```

```
Iteration 175 logL: -470.081298828125
Iteration 176 logL: -476.09564208984375
Iteration 177 logL: -466.91961669921875
Iteration 178 logL: -469.8760986328125
Iteration 179 logL: -451.9288330078125
Iteration 180 logL: -466.0950622558594
Iteration 181 logL: -462.870849609375
Iteration 182 logL: -444.9841613769531
Iteration 183 logL: -446.01702880859375
Iteration 184 logL: -447.80780029296875
Iteration 185 logL: -432.3081359863281
Iteration 186 logL: -438.62420654296875
Iteration 187 logL: -426.5700378417969
Iteration 188 logL: -426.06982421875
Iteration 189 logL: -414.849853515625
Iteration 190 logL: -420.2298583984375
Iteration 191 logL: -420.4183349609375
Iteration 192 logL: -417.06805419921875
Iteration 193 logL: -405.5826721191406
Iteration 194 logL: -401.24493408203125
Iteration 195 logL: -403.7947082519531
Iteration 196 logL: -400.0349426269531
Iteration 197 logL: -383.85406494140625
Iteration 198 logL: -394.892333984375
Iteration 199 logL: -388.74639892578125
Iteration 200 logL: -382.0386962890625
Iteration 201 logL: -376.4816589355469
Iteration 202 logL: -393.016845703125
Iteration 203 logL: -374.96209716796875
Iteration 204 logL: -364.32830810546875
Iteration 205 logL: -361.70440673828125
Iteration 206 logL: -355.5899658203125
Iteration 207 logL: -355.45263671875
Iteration 208 logL: -355.73486328125
Iteration 209 logL: -349.2461242675781
Iteration 210 logL: -359.9568176269531
Iteration 211 logL: -353.9207763671875
Iteration 212 logL: -342.2294921875
Iteration 213 logL: -326.1536865234375
Iteration 214 logL: -331.0500183105469
Iteration 215 logL: -330.4935302734375
Iteration 216 logL: -332.4503173828125
Iteration 217 logL: -323.48944091796875
Iteration 218 logL: -331.1150207519531
Iteration 219 logL: -337.6239013671875
Iteration 220 logL: -298.33416748046875
Iteration 221 logL: -320.0117492675781
Iteration 222 logL: -312.9443359375
Iteration 223 logL: -322.6186218261719
Iteration 224 logL: -299.63201904296875
Iteration 225 logL: -309.0345458984375
Iteration 226 logL: -297.46710205078125
Iteration 227 logL: -304.5844421386719
Iteration 228 logL: -296.8038330078125
Iteration 229 logL: -289.9383239746094
Iteration 230 logL: -273.607666015625
Iteration 231 logL: -273.5848388671875
Iteration 232 logL: -274.6050720214844
Iteration 233 logL: -279.795654296875
```

```
Iteration 234 logL: -265.371337890625
Iteration 235 logL: -268.1728210449219
Iteration 236 logL: -286.056640625
Iteration 237 logL: -258.6797180175781
Iteration 238 logL: -263.93536376953125
Iteration 239 logL: -246.6587371826172
Iteration 240 logL: -252.12295532226562
Iteration 241 logL: -238.30902099609375
Iteration 242 logL: -242.0396728515625
Iteration 243 logL: -278.215576171875
Iteration 244 logL: -252.17807006835938
Iteration 245 logL: -262.0987243652344
Iteration 246 logL: -233.15167236328125
Iteration 247 logL: -229.3694305419922
Iteration 248 logL: -235.25001525878906
Iteration 249 logL: -217.96432495117188
Iteration 250 logL: -253.81594848632812
Iteration 251 logL: -189.490966796875
Iteration 252 logL: -224.66537475585938
Iteration 253 logL: -199.2983856201172
Iteration 254 logL: -214.942626953125
Iteration 255 logL: -180.802734375
Iteration 256 logL: -218.85873413085938
Iteration 257 logL: -176.16651916503906
Iteration 258 logL: -189.5235137939453
Iteration 259 logL: -167.35398864746094
Iteration 260 logL: -180.61697387695312
Iteration 261 logL: -166.8021697998047
Iteration 262 logL: -156.7022705078125
Iteration 263 logL: -162.89773559570312
Iteration 264 logL: -166.24488830566406
Iteration 265 logL: -160.3469696044922
Iteration 266 logL: -153.28945922851562
Iteration 267 logL: -157.59815979003906
Iteration 268 logL: -167.51414489746094
Iteration 269 logL: -163.26901245117188
Iteration 270 logL: -139.74966430664062
Iteration 271 logL: -158.76559448242188
Iteration 272 logL: -141.02743530273438
Iteration 273 logL: -129.15756225585938
Iteration 274 logL: -145.99368286132812
Iteration 275 logL: -140.67652893066406
Iteration 276 logL: -113.24177551269531
Iteration 277 logL: -122.7694091796875
Iteration 278 logL: -106.48458099365234
Iteration 279 logL: -108.75043487548828
Iteration 280 logL: -91.87442016601562
Iteration 281 logL: -92.21498107910156
Iteration 282 logL: -100.6929702758789
Iteration 283 logL: -82.84451293945312
Iteration 284 logL: -92.75434875488281
Iteration 285 logL: -93.7781753540039
Iteration 286 logL: -91.920654296875
Iteration 287 logL: -86.06930541992188
Iteration 288 logL: -81.9096450805664
Iteration 289 logL: -70.30583953857422
Iteration 290 logL: -84.48768615722656
Iteration 291 logL: -71.4428939819336
Iteration 292 logL: -83.38542175292969
```

```
Iteration 293 logL: -96.51248931884766
Iteration 294 logL: -66.30433654785156
Iteration 295 logL: -73.40699005126953
Iteration 296 logL: -104.14588928222656
Iteration 297 logL: -83.77367401123047
Iteration 298 logL: -104.99046325683594
Iteration 299 logL: -85.58970642089844
Iteration 300 logL: -79.7596435546875
Iteration 301 logL: -53.35191345214844
Iteration 302 logL: -104.65110778808594
Iteration 303 logL: -61.388755798339844
Iteration 304 logL: -32.594459533691406
Iteration 305 logL: -37.874324798583984
Iteration 306 logL: -77.10919952392578
Iteration 307 logL: -33.217506408691406
Iteration 308 logL: -56.18412780761719
Iteration 309 logL: -87.82762908935547
Iteration 310 logL: -44.83682632446289
Iteration 311 logL: -18.37156105041504
Iteration 312 logL: -22.931732177734375
Iteration 313 logL: -24.70421600341797
Iteration 314 logL: -7.03718376159668
Iteration 315 logL: -18.170246124267578
Iteration 316 logL: -2.268517017364502
Iteration 317 logL: -17.25156021118164
Iteration 318 logL: -2.844649314880371
Iteration 319 logL: -19.26293182373047
Iteration 320 logL: -12.643939971923828
Iteration 321 logL: -1.202230453491211
Iteration 322 logL: 0.5165071487426758
Iteration 323 logL: 5.578118324279785
Iteration 324 logL: 23.714948654174805
Iteration 325 logL: 17.34128761291504
Iteration 326 logL: 31.618450164794922
Iteration 327 logL: 33.755428314208984
Iteration 328 logL: 42.324405670166016
Iteration 329 logL: 44.341514587402344
Iteration 330 logL: 46.069183349609375
Iteration 331 logL: 66.48011016845703
Iteration 332 logL: 58.236690521240234
Iteration 333 logL: 61.99576950073242
Iteration 334 logL: 34.77210998535156
Iteration 335 logL: 35.074771881103516
Iteration 336 logL: 66.44009399414062
Iteration 337 logL: 64.35301208496094
Iteration 338 logL: 59.575042724609375
Iteration 339 logL: 71.03413391113281
Iteration 340 logL: 55.05207824707031
Iteration 341 logL: 77.43887329101562
Iteration 342 logL: 56.923828125
Iteration 343 logL: 92.0802230834961
Iteration 344 logL: 70.06658172607422
Iteration 345 logL: 82.45234680175781
Iteration 346 logL: 73.00370788574219
Iteration 347 logL: 65.07442474365234
Iteration 348 logL: 73.87686157226562
Iteration 349 logL: 79.3005142211914
Iteration 350 logL: 55.198951721191406
Iteration 351 logL: 83.51309204101562
```

```
Iteration 352 logL: 96.02759552001953
Iteration 353 logL: 55.342220306396484
Iteration 354 logL: 89.10017395019531
Iteration 355 logL: 78.11026000976562
Iteration 356 logL: 82.8541488647461
Iteration 357 logL: 111.51866912841797
Iteration 358 logL: 67.80738830566406
Iteration 359 logL: 101.76077270507812
Iteration 360 logL: 77.43670654296875
Iteration 361 logL: 68.50812530517578
Iteration 362 logL: 124.23523712158203
Iteration 363 logL: 83.03801727294922
Iteration 364 logL: 106.09430694580078
Iteration 365 logL: 107.04542541503906
Iteration 366 logL: 114.77580261230469
Iteration 367 logL: 134.98370361328125
Iteration 368 logL: 110.24732971191406
Iteration 369 logL: 126.51893615722656
Iteration 370 logL: 128.8147430419922
Iteration 371 logL: 118.47773742675781
Iteration 372 logL: 108.15385437011719
Iteration 373 logL: 148.7581024169922
Iteration 374 logL: 144.29376220703125
Iteration 375 logL: 134.25433349609375
Iteration 376 logL: 146.80775451660156
Iteration 377 logL: 130.0442352294922
Iteration 378 logL: 141.30479431152344
Iteration 379 logL: 143.6103515625
Iteration 380 logL: 128.67282104492188
Iteration 381 logL: 137.861083984375
Iteration 382 logL: 161.84510803222656
Iteration 383 logL: 148.631103515625
Iteration 384 logL: 144.04946899414062
Iteration 385 logL: 140.3033447265625
Iteration 386 logL: 126.86595153808594
Iteration 387 logL: 136.71868896484375
Iteration 388 logL: 151.38510131835938
Iteration 389 logL: 150.2634735107422
Iteration 390 logL: 137.26632690429688
Iteration 391 logL: 163.96713256835938
Iteration 392 logL: 144.41830444335938
Iteration 393 logL: 153.94589233398438
Iteration 394 logL: 162.05921936035156
Iteration 395 logL: 188.280029296875
Iteration 396 logL: 181.39210510253906
Iteration 397 logL: 185.14877319335938
Iteration 398 logL: 143.304931640625
Iteration 399 logL: 193.80978393554688
Iteration 400 logL: 176.1383056640625
Iteration 401 logL: 160.65719604492188
Iteration 402 logL: 171.09719848632812
Iteration 403 logL: 200.96066284179688
Iteration 404 logL: 162.98590087890625
Iteration 405 logL: 176.97720336914062
Iteration 406 logL: 207.3131561279297
Iteration 407 logL: 175.38430786132812
Iteration 408 logL: 196.460693359375
Iteration 409 logL: 181.44216918945312
Iteration 410 logL: 175.09573364257812
```

```
Iteration 411 logL: 193.54617309570312
Iteration 412 logL: 162.65440368652344
Iteration 413 logL: 194.893798828125
Iteration 414 logL: 203.6279296875
Iteration 415 logL: 183.92750549316406
Iteration 416 logL: 228.21795654296875
Iteration 417 logL: 167.94210815429688
Iteration 418 logL: 187.89915466308594
Iteration 419 logL: 205.15663146972656
Iteration 420 logL: 150.40576171875
Iteration 421 logL: 170.99264526367188
Iteration 422 logL: 159.16787719726562
Iteration 423 logL: 212.30783081054688
Iteration 424 logL: 197.43667602539062
Iteration 425 logL: 209.94268798828125
Iteration 426 logL: 216.77255249023438
Iteration 427 logL: 178.6922607421875
Iteration 428 logL: 219.54080200195312
Iteration 429 logL: 204.65234375
Iteration 430 logL: 209.6220245361328
Iteration 431 logL: 187.70851135253906
Iteration 432 logL: 242.46409606933594
Iteration 433 logL: 210.56700134277344
Iteration 434 logL: 186.71800231933594
Iteration 435 logL: 225.4755401611328
Iteration 436 logL: 213.35824584960938
Iteration 437 logL: 202.34970092773438
Iteration 438 logL: 228.9705810546875
Iteration 439 logL: 198.14373779296875
Iteration 440 logL: 244.12088012695312
Iteration 441 logL: 236.96514892578125
Iteration 442 logL: 203.80783081054688
Iteration 443 logL: 218.20977783203125
Iteration 444 logL: 246.59051513671875
Iteration 445 logL: 243.72267150878906
Iteration 446 logL: 229.65682983398438
Iteration 447 logL: 250.5946044921875
Iteration 448 logL: 256.8073425292969
Iteration 449 logL: 256.71893310546875
Iteration 450 logL: 261.1754455566406
Iteration 451 logL: 244.5701904296875
Iteration 452 logL: 227.1672821044922
Iteration 453 logL: 264.9158020019531
Iteration 454 logL: 214.1546630859375
Iteration 455 logL: 226.58258056640625
Iteration 456 logL: 254.49893188476562
Iteration 457 logL: 209.53927612304688
Iteration 458 logL: 250.943115234375
Iteration 459 logL: 267.63482666015625
Iteration 460 logL: 270.44781494140625
Iteration 461 logL: 240.0543212890625
Iteration 462 logL: 272.0169677734375
Iteration 463 logL: 275.1575622558594
Iteration 464 logL: 253.59872436523438
Iteration 465 logL: 273.57080078125
Iteration 466 logL: 264.7949523925781
Iteration 467 logL: 274.9464111328125
Iteration 468 logL: 284.63385009765625
Iteration 469 logL: 266.6943664550781
```

```
Iteration 470 logL: 245.2529754638672
Iteration 471 logL: 249.99461364746094
Iteration 472 logL: 283.08782958984375
Iteration 473 logL: 289.62701416015625
Iteration 474 logL: 235.65704345703125
Iteration 475 logL: 271.97161865234375
Iteration 476 logL: 308.2992248535156
Iteration 477 logL: 241.93460083007812
Iteration 478 logL: 247.20474243164062
Iteration 479 logL: 294.2419738769531
Iteration 480 logL: 254.8139190673828
Iteration 481 logL: 273.0545959472656
Iteration 482 logL: 302.6460876464844
Iteration 483 logL: 276.9431457519531
Iteration 484 logL: 277.08782958984375
Iteration 485 logL: 280.70245361328125
Iteration 486 logL: 280.82159423828125
Iteration 487 logL: 284.2756042480469
Iteration 488 logL: 309.929931640625
Iteration 489 logL: 285.2984619140625
Iteration 490 logL: 267.9540710449219
Iteration 491 logL: 305.808837890625
Iteration 492 logL: 303.1220397949219
Iteration 493 logL: 255.30731201171875
Iteration 494 logL: 297.5326843261719
Iteration 495 logL: 311.49920654296875
Iteration 496 logL: 260.3905334472656
Iteration 497 logL: 296.09228515625
Iteration 498 logL: 317.2430419921875
Iteration 499 logL: 287.6947326660156
Iteration 500 logL: 301.8419189453125
Iteration 501 logL: 312.5766296386719
Iteration 502 logL: 299.15240478515625
Iteration 503 logL: 316.2951965332031
Iteration 504 logL: 299.87921142578125
Iteration 505 logL: 310.302978515625
Iteration 506 logL: 336.40557861328125
Iteration 507 logL: 292.2197265625
Iteration 508 logL: 327.706787109375
Iteration 509 logL: 285.19964599609375
Iteration 510 logL: 298.19598388671875
Iteration 511 logL: 333.88763427734375
Iteration 512 logL: 256.44342041015625
Iteration 513 logL: 321.1944580078125
Iteration 514 logL: 315.4454345703125
Iteration 515 logL: 323.385009765625
Iteration 516 logL: 341.30548095703125
Iteration 517 logL: 303.61865234375
Iteration 518 logL: 339.68682861328125
Iteration 519 logL: 310.6096496582031
Iteration 520 logL: 333.32598876953125
Iteration 521 logL: 348.69219970703125
Iteration 522 logL: 355.6949462890625
Iteration 523 logL: 327.2333068847656
Iteration 524 logL: 356.31658935546875
Iteration 525 logL: 333.55438232421875
Iteration 526 logL: 354.5309143066406
Iteration 527 logL: 322.62469482421875
Iteration 528 logL: 350.3382568359375
```

```
Iteration 529 logL: 340.6656494140625
Iteration 530 logL: 337.4777526855469
Iteration 531 logL: 349.881591796875
Iteration 532 logL: 352.5135803222656
Iteration 533 logL: 347.18756103515625
Iteration 534 logL: 356.7514343261719
Iteration 535 logL: 346.05859375
Iteration 536 logL: 348.28460693359375
Iteration 537 logL: 366.91717529296875
Iteration 538 logL: 346.76611328125
Iteration 539 logL: 348.06829833984375
Iteration 540 logL: 361.6518249511719
Iteration 541 logL: 334.48883056640625
Iteration 542 logL: 352.6024169921875
Iteration 543 logL: 367.4964599609375
Iteration 544 logL: 313.8736572265625
Iteration 545 logL: 349.51641845703125
Iteration 546 logL: 352.79718017578125
Iteration 547 logL: 341.87420654296875
Iteration 548 logL: 366.0464782714844
Iteration 549 logL: 342.61834716796875
Iteration 550 logL: 346.6181640625
Iteration 551 logL: 362.77764892578125
Iteration 552 logL: 320.08154296875
Iteration 553 logL: 273.3218078613281
Iteration 554 logL: 281.4428405761719
Iteration 555 logL: 306.0423583984375
Iteration 556 logL: 332.02569580078125
Iteration 557 logL: 285.3590087890625
Iteration 558 logL: 297.2322998046875
Iteration 559 logL: 371.57928466796875
Iteration 560 logL: 282.16912841796875
Iteration 561 logL: 307.2945556640625
Iteration 562 logL: 343.31304931640625
Iteration 563 logL: 362.9775390625
Iteration 564 logL: 351.9498291015625
Iteration 565 logL: 356.2451171875
Iteration 566 logL: 367.95294189453125
Iteration 567 logL: 384.5868225097656
Iteration 568 logL: 362.81201171875
Iteration 569 logL: 355.53350830078125
Iteration 570 logL: 399.5917663574219
Iteration 571 logL: 385.19622802734375
Iteration 572 logL: 364.68475341796875
Iteration 573 logL: 423.20477294921875
Iteration 574 logL: 350.1021728515625
Iteration 575 logL: 353.656982421875
Iteration 576 logL: 423.30731201171875
Iteration 577 logL: 329.29644775390625
Iteration 578 logL: 402.5065002441406
Iteration 579 logL: 401.5582275390625
Iteration 580 logL: 382.0030517578125
Iteration 581 logL: 396.90869140625
Iteration 582 logL: 419.322998046875
Iteration 583 logL: 431.70037841796875
Iteration 584 logL: 412.8690185546875
Iteration 585 logL: 432.121337890625
Iteration 586 logL: 434.9071044921875
Iteration 587 logL: 441.5890808105469
```

```
Iteration 588 logL: 451.06219482421875
Iteration 589 logL: 448.601806640625
Iteration 590 logL: 448.8460388183594
Iteration 591 logL: 429.44537353515625
Iteration 592 logL: 448.60113525390625
Iteration 593 logL: 449.3268127441406
Iteration 594 logL: 467.8370056152344
Iteration 595 logL: 437.68963623046875
Iteration 596 logL: 451.1627197265625
Iteration 597 logL: 443.6014709472656
Iteration 598 logL: 459.60467529296875
Iteration 599 logL: 434.41058349609375
Iteration 600 logL: 452.11810302734375
Iteration 601 logL: 377.40057373046875
Iteration 602 logL: 391.1588134765625
Iteration 603 logL: 409.9505615234375
Iteration 604 logL: 413.7198486328125
Iteration 605 logL: 388.3891296386719
Iteration 606 logL: 455.4639892578125
Iteration 607 logL: 470.5863342285156
Iteration 608 logL: 458.61444091796875
Iteration 609 logL: 469.2674255371094
Iteration 610 logL: 481.9045715332031
Iteration 611 logL: 440.2469482421875
Iteration 612 logL: 448.7007751464844
Iteration 613 logL: 461.82684326171875
Iteration 614 logL: 482.00531005859375
Iteration 615 logL: 466.93634033203125
Iteration 616 logL: 467.5984191894531
Iteration 617 logL: 452.7908935546875
Iteration 618 logL: 463.1468505859375
Iteration 619 logL: 472.0189514160156
Iteration 620 logL: 471.644287109375
Iteration 621 logL: 464.9411315917969
Iteration 622 logL: 461.953857421875
Iteration 623 logL: 462.7549133300781
Iteration 624 logL: 508.8385009765625
Iteration 625 logL: 472.0137023925781
Iteration 626 logL: 487.6414794921875
Iteration 627 logL: 486.8460998535156
Iteration 628 logL: 498.3939514160156
Iteration 629 logL: 509.3291015625
Iteration 630 logL: 471.84381103515625
Iteration 631 logL: 456.96502685546875
Iteration 632 logL: 443.15423583984375
Iteration 633 logL: 449.59222412109375
Iteration 634 logL: 442.419921875
Iteration 635 logL: 462.765625
Iteration 636 logL: 474.23779296875
Iteration 637 logL: 447.2813720703125
Iteration 638 logL: 368.62786865234375
Iteration 639 logL: 475.1927490234375
Iteration 640 logL: 493.5960693359375
Iteration 641 logL: 488.7607421875
Iteration 642 logL: 454.41827392578125
Iteration 643 logL: 389.7303161621094
Iteration 644 logL: 342.58612060546875
Iteration 645 logL: 450.5699462890625
Iteration 646 logL: 506.9656066894531
```

```
Iteration 647 logL: 427.39227294921875
Iteration 648 logL: 471.92926025390625
Iteration 649 logL: 506.56658935546875
Iteration 650 logL: 524.5393676757812
Iteration 651 logL: 531.684814453125
Iteration 652 logL: 533.3267822265625
Iteration 653 logL: 510.02490234375
Iteration 654 logL: 543.8906860351562
Iteration 655 logL: 514.4833984375
Iteration 656 logL: 507.345703125
Iteration 657 logL: 536.2933349609375
Iteration 658 logL: 549.052734375
Iteration 659 logL: 571.5628051757812
Iteration 660 logL: 552.716064453125
Iteration 661 logL: 557.7147216796875
Iteration 662 logL: 554.8577880859375
Iteration 663 logL: 466.18133544921875
Iteration 664 logL: 368.4473571777344
Iteration 665 logL: 360.48773193359375
Iteration 666 logL: 438.94110107421875
Iteration 667 logL: 572.2512817382812
Iteration 668 logL: 472.37847900390625
Iteration 669 logL: 530.16064453125
Iteration 670 logL: 553.6785888671875
Iteration 671 logL: 515.496337890625
Iteration 672 logL: 546.71826171875
Iteration 673 logL: 558.7097778320312
Iteration 674 logL: 536.4127197265625
Iteration 675 logL: 552.5570678710938
Iteration 676 logL: 563.292236328125
Iteration 677 logL: 493.20703125
Iteration 678 logL: 495.22198486328125
Iteration 679 logL: 554.2552490234375
Iteration 680 logL: 551.2715454101562
Iteration 681 logL: 563.9290771484375
Iteration 682 logL: 600.485107421875
Iteration 683 logL: 573.6139526367188
Iteration 684 logL: 579.658447265625
Iteration 685 logL: 611.6708984375
Iteration 686 logL: 602.7994995117188
Iteration 687 logL: 606.55029296875
Iteration 688 logL: 618.403076171875
Iteration 689 logL: 600.610107421875
Iteration 690 logL: 596.4420166015625
Iteration 691 logL: 602.8548583984375
Iteration 692 logL: 621.6720581054688
Iteration 693 logL: 622.0167236328125
Iteration 694 logL: 590.5845336914062
Iteration 695 logL: 587.8805541992188
Iteration 696 logL: 605.5408325195312
Iteration 697 logL: 605.4702758789062
Iteration 698 logL: 619.163330078125
Iteration 699 logL: 634.3576049804688
Iteration 700 logL: 628.318359375
Iteration 701 logL: 641.3948974609375
Iteration 702 logL: 643.2454833984375
Iteration 703 logL: 584.0679321289062
Iteration 704 logL: 617.7112426757812
Iteration 705 logL: 598.2427978515625
```

```
Iteration 706 logL: 587.2354736328125
Iteration 707 logL: 587.9380493164062
Iteration 708 logL: 623.57080078125
Iteration 709 logL: 615.5744018554688
Iteration 710 logL: 608.5718994140625
Iteration 711 logL: 649.4500122070312
Iteration 712 logL: 643.5366821289062
Iteration 713 logL: 602.8355102539062
Iteration 714 logL: 499.19744873046875
Iteration 715 logL: 320.66619873046875
Iteration 716 logL: 142.43173217773438
Iteration 717 logL: 293.0860900878906
Iteration 718 logL: 483.77728271484375
Iteration 719 logL: 385.66156005859375
Iteration 720 logL: 483.0570983886719
Iteration 721 logL: 426.65960693359375
Iteration 722 logL: 469.88397216796875
Iteration 723 logL: 506.4375915527344
Iteration 724 logL: 546.5360107421875
Iteration 725 logL: 518.2947998046875
Iteration 726 logL: 566.3784790039062
Iteration 727 logL: 566.865478515625
Iteration 728 logL: 552.1281127929688
Iteration 729 logL: 632.5340576171875
Iteration 730 logL: 582.7286376953125
Iteration 731 logL: 627.4791259765625
Iteration 732 logL: 657.6378784179688
Iteration 733 logL: 595.2085571289062
Iteration 734 logL: 644.544189453125
Iteration 735 logL: 670.85400390625
Iteration 736 logL: 644.2691650390625
Iteration 737 logL: 631.9349975585938
Iteration 738 logL: 683.2944946289062
Iteration 739 logL: 680.7939453125
Iteration 740 logL: 642.376220703125
Iteration 741 logL: 699.1393432617188
Iteration 742 logL: 672.81298828125
Iteration 743 logL: 710.6510620117188
Iteration 744 logL: 665.3779296875
Iteration 745 logL: 652.6602783203125
Iteration 746 logL: 706.9783325195312
Iteration 747 logL: 632.04052734375
Iteration 748 logL: 651.2955932617188
Iteration 749 logL: 712.1021728515625
Iteration 750 logL: 695.4866943359375
Iteration 751 logL: 700.0613403320312
Iteration 752 logL: 727.46826171875
Iteration 753 logL: 716.2306518554688
Iteration 754 logL: 735.77392578125
Iteration 755 logL: 719.989501953125
Iteration 756 logL: 698.1605834960938
Iteration 757 logL: 712.8348999023438
Iteration 758 logL: 714.6334228515625
Iteration 759 logL: 695.8239135742188
Iteration 760 logL: 721.2763671875
Iteration 761 logL: 685.1098022460938
Iteration 762 logL: 709.62255859375
Iteration 763 logL: 737.1724853515625
Iteration 764 logL: 682.0978393554688
```

```
Iteration 765 logL: 738.5803833007812
Iteration 766 logL: 743.95654296875
Iteration 767 logL: 743.0877685546875
Iteration 768 logL: 763.3662109375
Iteration 769 logL: 744.7777099609375
Iteration 770 logL: 769.8904418945312
Iteration 771 logL: 738.7501220703125
Iteration 772 logL: 746.0704956054688
Iteration 773 logL: 711.8949584960938
Iteration 774 logL: 725.2879028320312
Iteration 775 logL: 746.134521484375
Iteration 776 logL: 747.498291015625
Iteration 777 logL: 771.2698974609375
Iteration 778 logL: 767.6424560546875
Iteration 779 logL: 777.2642822265625
Iteration 780 logL: 784.9501342773438
Iteration 781 logL: 777.3809814453125
Iteration 782 logL: 794.6322021484375
Iteration 783 logL: 783.4398193359375
Iteration 784 logL: 778.979736328125
Iteration 785 logL: 764.1567993164062
Iteration 786 logL: 744.959228515625
Iteration 787 logL: 639.3988037109375
Iteration 788 logL: 491.92291259765625
Iteration 789 logL: 316.3533630371094
Iteration 790 logL: 404.31488037109375
Iteration 791 logL: 550.103515625
Iteration 792 logL: 449.9755859375
Iteration 793 logL: 521.9376220703125
Iteration 794 logL: 623.1883544921875
Iteration 795 logL: 559.3116455078125
Iteration 796 logL: 513.868896484375
Iteration 797 logL: 634.682373046875
Iteration 798 logL: 563.4610595703125
Iteration 799 logL: 525.4833984375
Iteration 800 logL: 700.590087890625
Iteration 801 logL: 613.9998779296875
Iteration 802 logL: 654.4163208007812
Iteration 803 logL: 728.7457275390625
Iteration 804 logL: 679.4075927734375
Iteration 805 logL: 765.2584228515625
Iteration 806 logL: 724.3031005859375
Iteration 807 logL: 746.6241455078125
Iteration 808 logL: 743.69873046875
Iteration 809 logL: 728.2290649414062
Iteration 810 logL: 754.6102294921875
Iteration 811 logL: 774.6488037109375
Iteration 812 logL: 755.5592651367188
Iteration 813 logL: 749.346923828125
Iteration 814 logL: 758.7463989257812
Iteration 815 logL: 763.64990234375
Iteration 816 logL: 757.1002197265625
Iteration 817 logL: 767.310302734375
Iteration 818 logL: 737.8364868164062
Iteration 819 logL: 805.5083618164062
Iteration 820 logL: 761.848388671875
Iteration 821 logL: 788.1103515625
Iteration 822 logL: 784.2445068359375
Iteration 823 logL: 800.9007568359375
```

```
Iteration 824 logL: 784.9432983398438
Iteration 825 logL: 800.532470703125
Iteration 826 logL: 796.6376953125
Iteration 827 logL: 777.4630126953125
Iteration 828 logL: 694.4915161132812
Iteration 829 logL: 752.05419921875
Iteration 830 logL: 830.629638671875
Iteration 831 logL: 782.41357421875
Iteration 832 logL: 725.2658081054688
Iteration 833 logL: 734.8521118164062
Iteration 834 logL: 811.8145141601562
Iteration 835 logL: 756.1964721679688
Iteration 836 logL: 684.70556640625
Iteration 837 logL: 803.984130859375
Iteration 838 logL: 795.5941162109375
Iteration 839 logL: 698.6575317382812
Iteration 840 logL: 760.0447998046875
Iteration 841 logL: 820.4157104492188
Iteration 842 logL: 753.5838012695312
Iteration 843 logL: 727.7915649414062
Iteration 844 logL: 860.5098876953125
Iteration 845 logL: 752.8709716796875
Iteration 846 logL: 721.5332641601562
Iteration 847 logL: 814.689453125
Iteration 848 logL: 696.8302001953125
Iteration 849 logL: 671.3468017578125
Iteration 850 logL: 831.80078125
Iteration 851 logL: 762.1656494140625
Iteration 852 logL: 782.39697265625
Iteration 853 logL: 810.5587158203125
Iteration 854 logL: 791.9981079101562
Iteration 855 logL: 851.9354248046875
Iteration 856 logL: 827.7073974609375
Iteration 857 logL: 819.2796630859375
Iteration 858 logL: 856.8084716796875
Iteration 859 logL: 845.40234375
Iteration 860 logL: 791.4742431640625
Iteration 861 logL: 826.3485107421875
Iteration 862 logL: 832.392578125
Iteration 863 logL: 794.6221923828125
Iteration 864 logL: 788.262451171875
Iteration 865 logL: 837.7229614257812
Iteration 866 logL: 832.43017578125
Iteration 867 logL: 823.0416259765625
Iteration 868 logL: 891.2208251953125
Iteration 869 logL: 815.0733642578125
Iteration 870 logL: 781.8444213867188
Iteration 871 logL: 833.4432983398438
Iteration 872 logL: 825.365966796875
Iteration 873 logL: 808.8866577148438
Iteration 874 logL: 861.6322021484375
Iteration 875 logL: 813.296142578125
Iteration 876 logL: 750.08056640625
Iteration 877 logL: 772.3712158203125
Iteration 878 logL: 872.7282104492188
Iteration 879 logL: 798.1844482421875
Iteration 880 logL: 766.4964599609375
Iteration 881 logL: 780.685302734375
Iteration 882 logL: 848.6764526367188
```

```
Iteration 883 logL: 856.498779296875
Iteration 884 logL: 808.150146484375
Iteration 885 logL: 817.8626708984375
Iteration 886 logL: 867.3085327148438
Iteration 887 logL: 838.1316528320312
Iteration 888 logL: 844.7845458984375
Iteration 889 logL: 887.7886352539062
Iteration 890 logL: 868.22265625
Iteration 891 logL: 813.8795166015625
Iteration 892 logL: 796.704345703125
Iteration 893 logL: 793.932861328125
Iteration 894 logL: 867.7857666015625
Iteration 895 logL: 856.1488037109375
Iteration 896 logL: 843.4646606445312
Iteration 897 logL: 868.587890625
Iteration 898 logL: 854.2940673828125
Iteration 899 logL: 876.6920776367188
Iteration 900 logL: 873.056396484375
Iteration 901 logL: 831.32470703125
Iteration 902 logL: 838.5699462890625
Iteration 903 logL: 830.3270874023438
Iteration 904 logL: 802.0255126953125
Iteration 905 logL: 809.5960693359375
Iteration 906 logL: 822.2182006835938
Iteration 907 logL: 873.634765625
Iteration 908 logL: 875.795166015625
Iteration 909 logL: 862.3692626953125
Iteration 910 logL: 850.43310546875
Iteration 911 logL: 833.352294921875
Iteration 912 logL: 849.2166748046875
Iteration 913 logL: 902.6884765625
Iteration 914 logL: 877.7923583984375
Iteration 915 logL: 834.2813110351562
Iteration 916 logL: 890.005126953125
Iteration 917 logL: 871.7969970703125
Iteration 918 logL: 819.1071166992188
Iteration 919 logL: 815.7681884765625
Iteration 920 logL: 830.6129150390625
Iteration 921 logL: 830.6146240234375
Iteration 922 logL: 858.05615234375
Iteration 923 logL: 909.9656982421875
Iteration 924 logL: 904.9473876953125
Iteration 925 logL: 869.4320678710938
Iteration 926 logL: 858.7607421875
Iteration 927 logL: 860.7669677734375
Iteration 928 logL: 899.0844116210938
Iteration 929 logL: 860.9967041015625
Iteration 930 logL: 841.0380859375
Iteration 931 logL: 819.872314453125
Iteration 932 logL: 874.8673095703125
Iteration 933 logL: 879.4820556640625
Iteration 934 logL: 929.922607421875
Iteration 935 logL: 860.5730590820312
Iteration 936 logL: 819.739013671875
Iteration 937 logL: 815.633544921875
Iteration 938 logL: 818.7364501953125
Iteration 939 logL: 887.1773681640625
Iteration 940 logL: 875.120361328125
Iteration 941 logL: 792.4935302734375
```

```
Iteration 942 logL: 793.6029663085938
Iteration 943 logL: 878.1333618164062
Iteration 944 logL: 839.19091796875
Iteration 945 logL: 752.7464599609375
Iteration 946 logL: 837.2781982421875
Iteration 947 logL: 889.81591796875
Iteration 948 logL: 903.303955078125
Iteration 949 logL: 825.2437744140625
Iteration 950 logL: 791.29443359375
Iteration 951 logL: 855.7703857421875
Iteration 952 logL: 902.7705078125
Iteration 953 logL: 875.7179565429688
Iteration 954 logL: 825.7620239257812
Iteration 955 logL: 852.2694702148438
Iteration 956 logL: 845.295166015625
Iteration 957 logL: 781.8325805664062
Iteration 958 logL: 892.7081298828125
Iteration 959 logL: 901.98876953125
Iteration 960 logL: 898.4300537109375
Iteration 961 logL: 904.5157470703125
Iteration 962 logL: 887.3248901367188
Iteration 963 logL: 866.4799194335938
Iteration 964 logL: 893.6541748046875
Iteration 965 logL: 920.5748901367188
Iteration 966 logL: 902.6015625
Iteration 967 logL: 904.662353515625
Iteration 968 logL: 919.341064453125
Iteration 969 logL: 909.395751953125
Iteration 970 logL: 870.550537109375
Iteration 971 logL: 893.7337036132812
Iteration 972 logL: 910.2001953125
Iteration 973 logL: 924.77978515625
Iteration 974 logL: 902.1597900390625
Iteration 975 logL: 934.772705078125
Iteration 976 logL: 943.2953491210938
Iteration 977 logL: 957.3778076171875
Iteration 978 logL: 941.6713256835938
Iteration 979 logL: 926.445068359375
Iteration 980 logL: 856.239501953125
Iteration 981 logL: 803.87353515625
Iteration 982 logL: 738.852294921875
Iteration 983 logL: 812.7286987304688
Iteration 984 logL: 926.26708984375
Iteration 985 logL: 815.5946655273438
Iteration 986 logL: 769.1473388671875
Iteration 987 logL: 781.5574951171875
Iteration 988 logL: 847.5048828125
Iteration 989 logL: 873.23681640625
Iteration 990 logL: 863.3602905273438
Iteration 991 logL: 855.444580078125
Iteration 992 logL: 900.2230224609375
Iteration 993 logL: 901.166259765625
Iteration 994 logL: 880.4569702148438
Iteration 995 logL: 845.4869384765625
Iteration 996 logL: 926.9755859375
Iteration 997 logL: 896.4877319335938
Iteration 998 logL: 858.26904296875
Iteration 999 logL: 819.2239990234375
Iteration 1000 logL: 780.9205322265625
```

```
Iteration 1001 logL: 768.3468017578125
Iteration 1002 logL: 857.719482421875
Iteration 1003 logL: 881.823974609375
Iteration 1004 logL: 868.834228515625
Iteration 1005 logL: 911.712646484375
Iteration 1006 logL: 916.9375
Iteration 1007 logL: 890.4544677734375
Iteration 1008 logL: 876.24267578125
Iteration 1009 logL: 924.3963623046875
Iteration 1010 logL: 865.1298217773438
Iteration 1011 logL: 862.833984375
Iteration 1012 logL: 944.04248046875
Iteration 1013 logL: 892.9779663085938
Iteration 1014 logL: 868.5478515625
Iteration 1015 logL: 944.477294921875
Iteration 1016 logL: 945.9141845703125
Iteration 1017 logL: 920.2564697265625
Iteration 1018 logL: 919.82080078125
Iteration 1019 logL: 940.7965087890625
Iteration 1020 logL: 966.6114501953125
Iteration 1021 logL: 954.10205078125
Iteration 1022 logL: 910.573486328125
Iteration 1023 logL: 893.2075805664062
Iteration 1024 logL: 912.8754272460938
Iteration 1025 logL: 964.9837646484375
Iteration 1026 logL: 982.4212646484375
Iteration 1027 logL: 947.7470092773438
Iteration 1028 logL: 941.65869140625
Iteration 1029 logL: 964.0009155273438
Iteration 1030 logL: 916.5143432617188
Iteration 1031 logL: 911.6053466796875
Iteration 1032 logL: 957.221435546875
Iteration 1033 logL: 965.1920166015625
Iteration 1034 logL: 967.8538818359375
Iteration 1035 logL: 920.168701171875
Iteration 1036 logL: 953.2971801757812
Iteration 1037 logL: 986.2423095703125
Iteration 1038 logL: 921.7342529296875
Iteration 1039 logL: 852.3580322265625
Iteration 1040 logL: 922.000732421875
Iteration 1041 logL: 982.525634765625
Iteration 1042 logL: 870.6652221679688
Iteration 1043 logL: 803.65234375
Iteration 1044 logL: 864.9774169921875
Iteration 1045 logL: 901.2490234375
Iteration 1046 logL: 903.4307861328125
Iteration 1047 logL: 865.6826782226562
Iteration 1048 logL: 931.6357421875
Iteration 1049 logL: 945.572265625
Iteration 1050 logL: 905.680419921875
Iteration 1051 logL: 926.7242431640625
Iteration 1052 logL: 947.9833984375
Iteration 1053 logL: 874.84130859375
Iteration 1054 logL: 910.8488159179688
Iteration 1055 logL: 921.4390258789062
Iteration 1056 logL: 900.1387329101562
Iteration 1057 logL: 919.57666015625
Iteration 1058 logL: 962.814208984375
Iteration 1059 logL: 959.2972412109375
```

```
Iteration 1060 logL: 928.4442749023438
Iteration 1061 logL: 875.626953125
Iteration 1062 logL: 946.3865966796875
Iteration 1063 logL: 971.6014404296875
Iteration 1064 logL: 982.6182861328125
Iteration 1065 logL: 946.5263671875
Iteration 1066 logL: 918.4513549804688
Iteration 1067 logL: 965.3938598632812
Iteration 1068 logL: 907.2666015625
Iteration 1069 logL: 925.3719482421875
Iteration 1070 logL: 930.8196411132812
Iteration 1071 logL: 937.8133544921875
Iteration 1072 logL: 967.342529296875
Iteration 1073 logL: 950.20068359375
Iteration 1074 logL: 1000.607177734375
Iteration 1075 logL: 941.8753662109375
Iteration 1076 logL: 960.178955078125
Iteration 1077 logL: 944.3099365234375
Iteration 1078 logL: 985.7518310546875
Iteration 1079 logL: 951.30078125
Iteration 1080 logL: 926.48828125
Iteration 1081 logL: 946.25244140625
Iteration 1082 logL: 906.4581909179688
Iteration 1083 logL: 961.4539184570312
Iteration 1084 logL: 994.9761962890625
Iteration 1085 logL: 912.68408203125
Iteration 1086 logL: 841.2335205078125
Iteration 1087 logL: 915.2471923828125
Iteration 1088 logL: 978.2227783203125
Iteration 1089 logL: 903.20703125
Iteration 1090 logL: 759.9452514648438
Iteration 1091 logL: 850.0844116210938
Iteration 1092 logL: 932.9766235351562
Iteration 1093 logL: 958.7154541015625
Iteration 1094 logL: 891.2054443359375
Iteration 1095 logL: 940.0362548828125
Iteration 1096 logL: 966.9674072265625
Iteration 1097 logL: 935.2928466796875
Iteration 1098 logL: 921.0347900390625
Iteration 1099 logL: 955.9818115234375
Iteration 1100 logL: 978.80078125
Iteration 1101 logL: 974.4541015625
Iteration 1102 logL: 996.4125366210938
Iteration 1103 logL: 981.74365234375
Iteration 1104 logL: 1009.5087890625
Iteration 1105 logL: 971.9537353515625
Iteration 1106 logL: 964.0491943359375
Iteration 1107 logL: 1026.899169921875
Iteration 1108 logL: 995.9063720703125
Iteration 1109 logL: 957.7525634765625
Iteration 1110 logL: 1011.7197265625
Iteration 1111 logL: 962.6839599609375
Iteration 1112 logL: 881.141357421875
Iteration 1113 logL: 860.9453125
Iteration 1114 logL: 998.08642578125
Iteration 1115 logL: 1001.9853515625
Iteration 1116 logL: 884.92822265625
Iteration 1117 logL: 877.4230346679688
Iteration 1118 logL: 998.454345703125
```

```
Iteration 1119 logL: 1001.0167236328125
Iteration 1120 logL: 936.878662109375
Iteration 1121 logL: 1004.7036743164062
Iteration 1122 logL: 976.13134765625
Iteration 1123 logL: 909.76953125
Iteration 1124 logL: 954.080322265625
Iteration 1125 logL: 1003.7349243164062
Iteration 1126 logL: 1011.4635620117188
Iteration 1127 logL: 938.172607421875
Iteration 1128 logL: 990.42236328125
Iteration 1129 logL: 999.8948974609375
Iteration 1130 logL: 981.76708984375
Iteration 1131 logL: 958.307861328125
Iteration 1132 logL: 1012.425048828125
Iteration 1133 logL: 1025.4129638671875
Iteration 1134 logL: 994.3681640625
Iteration 1135 logL: 1007.59375
Iteration 1136 logL: 1030.0966796875
Iteration 1137 logL: 947.695556640625
Iteration 1138 logL: 958.6990966796875
Iteration 1139 logL: 1056.02197265625
Iteration 1140 logL: 980.3312377929688
Iteration 1141 logL: 976.8612060546875
Iteration 1142 logL: 1042.078125
Iteration 1143 logL: 987.259033203125
Iteration 1144 logL: 964.2059936523438
Iteration 1145 logL: 1028.74853515625
Iteration 1146 logL: 965.8903198242188
Iteration 1147 logL: 960.841064453125
Iteration 1148 logL: 1011.68798828125
Iteration 1149 logL: 1034.32861328125
Iteration 1150 logL: 1013.7468872070312
Iteration 1151 logL: 1002.2064208984375
Iteration 1152 logL: 1006.8956298828125
Iteration 1153 logL: 964.54345703125
Iteration 1154 logL: 1031.6063232421875
Iteration 1155 logL: 1049.6331787109375
Iteration 1156 logL: 1019.5391235351562
Iteration 1157 logL: 1055.212158203125
Iteration 1158 logL: 1051.8448486328125
Iteration 1159 logL: 1052.8662109375
Iteration 1160 logL: 1039.8751220703125
Iteration 1161 logL: 1030.601806640625
Iteration 1162 logL: 1031.4415283203125
Iteration 1163 logL: 995.48974609375
Iteration 1164 logL: 998.2366943359375
Iteration 1165 logL: 1006.3999633789062
Iteration 1166 logL: 1016.656494140625
Iteration 1167 logL: 1008.775146484375
Iteration 1168 logL: 963.3741455078125
Iteration 1169 logL: 977.933349609375
Iteration 1170 logL: 1023.3839721679688
Iteration 1171 logL: 989.4598388671875
Iteration 1172 logL: 879.897705078125
Iteration 1173 logL: 811.1166381835938
Iteration 1174 logL: 878.1060180664062
Iteration 1175 logL: 989.59619140625
Iteration 1176 logL: 1031.4664306640625
Iteration 1177 logL: 992.6622314453125
```

```
Iteration 1178 logL: 944.945068359375
Iteration 1179 logL: 921.537109375
Iteration 1180 logL: 977.43359375
Iteration 1181 logL: 1006.498046875
Iteration 1182 logL: 985.47509765625
Iteration 1183 logL: 1020.143798828125
Iteration 1184 logL: 1044.1263427734375
Iteration 1185 logL: 998.0515747070312
Iteration 1186 logL: 956.2380981445312
Iteration 1187 logL: 1031.897705078125
Iteration 1188 logL: 1041.5458984375
Iteration 1189 logL: 978.9465942382812
Iteration 1190 logL: 991.716064453125
Iteration 1191 logL: 977.06689453125
Iteration 1192 logL: 934.6519775390625
Iteration 1193 logL: 967.3858642578125
Iteration 1194 logL: 1064.3514404296875
Iteration 1195 logL: 994.1834716796875
Iteration 1196 logL: 913.5711669921875
Iteration 1197 logL: 1038.669189453125
Iteration 1198 logL: 1051.0638427734375
Iteration 1199 logL: 1012.953125
Iteration 1200 logL: 1051.5821533203125
Iteration 1201 logL: 1011.1603393554688
Iteration 1202 logL: 986.8899536132812
Iteration 1203 logL: 1060.231689453125
Iteration 1204 logL: 1048.78955078125
Iteration 1205 logL: 1031.1334228515625
Iteration 1206 logL: 1015.703125
Iteration 1207 logL: 1003.51708984375
Iteration 1208 logL: 998.8729248046875
Iteration 1209 logL: 1008.5675048828125
Iteration 1210 logL: 1049.3314208984375
Iteration 1211 logL: 1019.5850830078125
Iteration 1212 logL: 951.7098388671875
Iteration 1213 logL: 995.2652587890625
Iteration 1214 logL: 1004.86474609375
Iteration 1215 logL: 934.3072509765625
Iteration 1216 logL: 995.8787841796875
Iteration 1217 logL: 1037.0078125
Iteration 1218 logL: 1040.4296875
Iteration 1219 logL: 1073.4080810546875
Iteration 1220 logL: 1036.883056640625
Iteration 1221 logL: 1020.2435302734375
Iteration 1222 logL: 1039.113037109375
Iteration 1223 logL: 1084.9283447265625
Iteration 1224 logL: 1020.710693359375
Iteration 1225 logL: 1017.2459716796875
Iteration 1226 logL: 1056.23681640625
Iteration 1227 logL: 1081.697509765625
Iteration 1228 logL: 1038.1768798828125
Iteration 1229 logL: 1006.4008178710938
Iteration 1230 logL: 1024.9195556640625
Iteration 1231 logL: 1004.7852172851562
Iteration 1232 logL: 1003.697021484375
Iteration 1233 logL: 1050.723388671875
Iteration 1234 logL: 1042.200439453125
Iteration 1235 logL: 1068.2484130859375
Iteration 1236 logL: 1042.873291015625
```

```
Iteration 1237 logL: 1020.478515625
Iteration 1238 logL: 1056.9447021484375
Iteration 1239 logL: 976.2313232421875
Iteration 1240 logL: 971.6300048828125
Iteration 1241 logL: 1030.937744140625
Iteration 1242 logL: 959.7384033203125
Iteration 1243 logL: 906.636962890625
Iteration 1244 logL: 955.5054321289062
Iteration 1245 logL: 1034.653564453125
Iteration 1246 logL: 1024.2708740234375
Iteration 1247 logL: 1026.6654052734375
Iteration 1248 logL: 1050.3961181640625
Iteration 1249 logL: 1088.3502197265625
Iteration 1250 logL: 1010.0401000976562
Iteration 1251 logL: 983.81884765625
Iteration 1252 logL: 1063.0697021484375
Iteration 1253 logL: 1056.481689453125
Iteration 1254 logL: 981.06201171875
Iteration 1255 logL: 908.89111328125
Iteration 1256 logL: 967.3739013671875
Iteration 1257 logL: 956.0880126953125
Iteration 1258 logL: 944.5364990234375
Iteration 1259 logL: 1008.58984375
Iteration 1260 logL: 1006.234375
Iteration 1261 logL: 926.1055297851562
Iteration 1262 logL: 994.1201171875
Iteration 1263 logL: 1010.45263671875
Iteration 1264 logL: 1008.4803466796875
Iteration 1265 logL: 1023.6510009765625
Iteration 1266 logL: 1062.322265625
Iteration 1267 logL: 977.5034790039062
Iteration 1268 logL: 996.441162109375
Iteration 1269 logL: 1047.262451171875
Iteration 1270 logL: 1019.5039672851562
Iteration 1271 logL: 1025.252197265625
Iteration 1272 logL: 1055.385986328125
Iteration 1273 logL: 1031.588623046875
Iteration 1274 logL: 1107.1390380859375
Iteration 1275 logL: 1040.4254150390625
Iteration 1276 logL: 1023.6982421875
Iteration 1277 logL: 1022.4998779296875
Iteration 1278 logL: 1074.883056640625
Iteration 1279 logL: 1070.1644287109375
Iteration 1280 logL: 982.242919921875
Iteration 1281 logL: 1027.20166015625
Iteration 1282 logL: 1089.7889404296875
Iteration 1283 logL: 1001.1393432617188
Iteration 1284 logL: 922.6226806640625
Iteration 1285 logL: 994.6102294921875
Iteration 1286 logL: 1070.2705078125
Iteration 1287 logL: 936.4522705078125
Iteration 1288 logL: 910.5323486328125
Iteration 1289 logL: 1065.896728515625
Iteration 1290 logL: 1045.0638427734375
Iteration 1291 logL: 1026.94287109375
Iteration 1292 logL: 1060.2962646484375
Iteration 1293 logL: 1088.6358642578125
Iteration 1294 logL: 1059.0067138671875
Iteration 1295 logL: 1051.98876953125
```

```
Iteration 1296 logL: 1108.985107421875
Iteration 1297 logL: 1087.4686279296875
Iteration 1298 logL: 1111.777587890625
Iteration 1299 logL: 1076.643798828125
Iteration 1300 logL: 1063.34326171875
Iteration 1301 logL: 1057.577392578125
Iteration 1302 logL: 1099.7952880859375
Iteration 1303 logL: 1119.310302734375
Iteration 1304 logL: 1088.662841796875
Iteration 1305 logL: 1066.4080810546875
Iteration 1306 logL: 1066.353515625
Iteration 1307 logL: 1056.935302734375
Iteration 1308 logL: 1099.0728759765625
Iteration 1309 logL: 1112.1446533203125
Iteration 1310 logL: 1099.04638671875
Iteration 1311 logL: 1104.18505859375
Iteration 1312 logL: 1078.65576171875
Iteration 1313 logL: 1052.7510986328125
Iteration 1314 logL: 1078.056396484375
Iteration 1315 logL: 1085.4930419921875
Iteration 1316 logL: 1096.273193359375
Iteration 1317 logL: 1052.4189453125
Iteration 1318 logL: 1045.9833984375
Iteration 1319 logL: 1096.1787109375
Iteration 1320 logL: 1106.799072265625
Iteration 1321 logL: 1083.4617919921875
Iteration 1322 logL: 1083.7730712890625
Iteration 1323 logL: 1087.7950439453125
Iteration 1324 logL: 1128.942138671875
Iteration 1325 logL: 1075.8956298828125
Iteration 1326 logL: 1087.619873046875
Iteration 1327 logL: 1092.378173828125
Iteration 1328 logL: 1075.720458984375
Iteration 1329 logL: 1060.21875
Iteration 1330 logL: 1051.198974609375
Iteration 1331 logL: 1042.868408203125
Iteration 1332 logL: 965.0197143554688
Iteration 1333 logL: 1022.3446655273438
Iteration 1334 logL: 1112.038330078125
Iteration 1335 logL: 1114.69189453125
Iteration 1336 logL: 1079.322509765625
Iteration 1337 logL: 1043.923583984375
Iteration 1338 logL: 1049.371826171875
Iteration 1339 logL: 1058.91015625
Iteration 1340 logL: 1086.10595703125
Iteration 1341 logL: 1026.8619384765625
Iteration 1342 logL: 1050.6318359375
Iteration 1343 logL: 1062.94970703125
Iteration 1344 logL: 1031.307861328125
Iteration 1345 logL: 980.0872802734375
Iteration 1346 logL: 1037.4945068359375
Iteration 1347 logL: 1095.67236328125
Iteration 1348 logL: 1066.962158203125
Iteration 1349 logL: 1024.1964111328125
Iteration 1350 logL: 1004.3167114257812
Iteration 1351 logL: 1067.9764404296875
Iteration 1352 logL: 1083.7701416015625
Iteration 1353 logL: 1068.117431640625
Iteration 1354 logL: 1092.052978515625
```

```
Iteration 1355 logL: 1078.3023681640625
Iteration 1356 logL: 1075.7003173828125
Iteration 1357 logL: 1041.724853515625
Iteration 1358 logL: 1061.9017333984375
Iteration 1359 logL: 1059.269287109375
Iteration 1360 logL: 1093.462158203125
Iteration 1361 logL: 1111.3243408203125
Iteration 1362 logL: 1106.2779541015625
Iteration 1363 logL: 1117.1756591796875
Iteration 1364 logL: 1064.3497314453125
Iteration 1365 logL: 1056.778076171875
Iteration 1366 logL: 1048.870361328125
Iteration 1367 logL: 1084.9849853515625
Iteration 1368 logL: 1124.86474609375
Iteration 1369 logL: 1121.9947509765625
Iteration 1370 logL: 1099.089599609375
Iteration 1371 logL: 1035.025634765625
Iteration 1372 logL: 1023.792724609375
Iteration 1373 logL: 1059.32421875
Iteration 1374 logL: 1106.70947265625
Iteration 1375 logL: 1108.5257568359375
Iteration 1376 logL: 1032.7432861328125
Iteration 1377 logL: 1028.4566650390625
Iteration 1378 logL: 1040.384033203125
Iteration 1379 logL: 1082.500732421875
Iteration 1380 logL: 1082.476806640625
Iteration 1381 logL: 1092.5838623046875
Iteration 1382 logL: 1105.9150390625
Iteration 1383 logL: 1055.1463623046875
Iteration 1384 logL: 1038.2843017578125
Iteration 1385 logL: 1018.0367431640625
Iteration 1386 logL: 1111.154052734375
Iteration 1387 logL: 1075.710205078125
Iteration 1388 logL: 982.443603515625
Iteration 1389 logL: 1020.1004638671875
Iteration 1390 logL: 1092.765380859375
Iteration 1391 logL: 1062.5311279296875
Iteration 1392 logL: 957.3757934570312
Iteration 1393 logL: 927.8474731445312
Iteration 1394 logL: 1021.0423583984375
Iteration 1395 logL: 1045.3433837890625
Iteration 1396 logL: 1009.72314453125
Iteration 1397 logL: 991.603759765625
Iteration 1398 logL: 977.8209228515625
Iteration 1399 logL: 1044.0107421875
Iteration 1400 logL: 1090.32763671875
Iteration 1401 logL: 1014.3770751953125
Iteration 1402 logL: 1045.04443359375
Iteration 1403 logL: 1091.310546875
Iteration 1404 logL: 1089.251708984375
Iteration 1405 logL: 1031.6688232421875
Iteration 1406 logL: 1103.1978759765625
Iteration 1407 logL: 1049.5902099609375
Iteration 1408 logL: 972.9288330078125
Iteration 1409 logL: 1061.5770263671875
Iteration 1410 logL: 1120.45458984375
Iteration 1411 logL: 1070.0792236328125
Iteration 1412 logL: 1071.098876953125
Iteration 1413 logL: 1112.0489501953125
```

```
Iteration 1414 logL: 1115.5775146484375
Iteration 1415 logL: 1103.9989013671875
Iteration 1416 logL: 1116.6826171875
Iteration 1417 logL: 1133.7315673828125
Iteration 1418 logL: 1117.7760009765625
Iteration 1419 logL: 1141.715087890625
Iteration 1420 logL: 1124.56689453125
Iteration 1421 logL: 1122.458984375
Iteration 1422 logL: 1108.280029296875
Iteration 1423 logL: 1149.3221435546875
Iteration 1424 logL: 1111.755615234375
Iteration 1425 logL: 1128.42724609375
Iteration 1426 logL: 1135.68798828125
Iteration 1427 logL: 1094.2342529296875
Iteration 1428 logL: 1107.726806640625
Iteration 1429 logL: 1115.798095703125
Iteration 1430 logL: 1131.3902587890625
Iteration 1431 logL: 1063.142578125
Iteration 1432 logL: 1103.74658203125
Iteration 1433 logL: 1121.95703125
Iteration 1434 logL: 1134.5574951171875
Iteration 1435 logL: 1055.442138671875
Iteration 1436 logL: 1040.49755859375
Iteration 1437 logL: 1087.38525390625
Iteration 1438 logL: 1128.4381103515625
Iteration 1439 logL: 1031.7076416015625
Iteration 1440 logL: 1049.7083740234375
Iteration 1441 logL: 1073.883056640625
Iteration 1442 logL: 1119.4761962890625
Iteration 1443 logL: 1102.576416015625
Iteration 1444 logL: 1076.0848388671875
Iteration 1445 logL: 1096.4810791015625
Iteration 1446 logL: 1128.7352294921875
Iteration 1447 logL: 1090.688232421875
Iteration 1448 logL: 1072.9058837890625
Iteration 1449 logL: 1128.2227783203125
Iteration 1450 logL: 1124.782470703125
Iteration 1451 logL: 1063.9635009765625
Iteration 1452 logL: 1107.669189453125
Iteration 1453 logL: 1123.579833984375
Iteration 1454 logL: 1116.1767578125
Iteration 1455 logL: 1110.367919921875
Iteration 1456 logL: 1111.6942138671875
Iteration 1457 logL: 1083.7950439453125
Iteration 1458 logL: 1094.4737548828125
Iteration 1459 logL: 1155.8670654296875
Iteration 1460 logL: 1086.921142578125
Iteration 1461 logL: 1065.3837890625
Iteration 1462 logL: 1097.5452880859375
Iteration 1463 logL: 1109.773193359375
Iteration 1464 logL: 1126.924072265625
Iteration 1465 logL: 1154.6607666015625
Iteration 1466 logL: 1095.8040771484375
Iteration 1467 logL: 1116.720703125
Iteration 1468 logL: 1137.73388671875
Iteration 1469 logL: 1091.8045654296875
Iteration 1470 logL: 1142.22705078125
Iteration 1471 logL: 1161.355224609375
Iteration 1472 logL: 1116.193359375
```

```
Iteration 1473 logL: 1105.38232421875
Iteration 1474 logL: 1097.63232421875
Iteration 1475 logL: 1075.0294189453125
Iteration 1476 logL: 1101.552978515625
Iteration 1477 logL: 1087.760986328125
Iteration 1478 logL: 1111.390869140625
Iteration 1479 logL: 1136.67431640625
Iteration 1480 logL: 1082.6630859375
Iteration 1481 logL: 1000.4555053710938
Iteration 1482 logL: 988.0201416015625
Iteration 1483 logL: 1026.737060546875
Iteration 1484 logL: 1102.697998046875
Iteration 1485 logL: 1126.57958984375
Iteration 1486 logL: 987.6395263671875
Iteration 1487 logL: 942.8388671875
Iteration 1488 logL: 1059.6405029296875
Iteration 1489 logL: 1120.0518798828125
Iteration 1490 logL: 1008.0853271484375
Iteration 1491 logL: 981.39990234375
Iteration 1492 logL: 1108.1512451171875
Iteration 1493 logL: 1122.833740234375
Iteration 1494 logL: 1100.7647705078125
Iteration 1495 logL: 1124.036376953125
Iteration 1496 logL: 1084.322265625
Iteration 1497 logL: 1044.4805908203125
Iteration 1498 logL: 1078.3441162109375
Iteration 1499 logL: 1157.02294921875
Iteration 1500 logL: 1100.0927734375
Iteration 1501 logL: 998.154296875
Iteration 1502 logL: 1075.4093017578125
Iteration 1503 logL: 1142.6153564453125
Iteration 1504 logL: 1091.431396484375
Iteration 1505 logL: 1107.06396484375
Iteration 1506 logL: 1132.876708984375
Iteration 1507 logL: 1128.831298828125
Iteration 1508 logL: 1117.4716796875
Iteration 1509 logL: 1155.6171875
Iteration 1510 logL: 1137.26220703125
Iteration 1511 logL: 1081.8626708984375
Iteration 1512 logL: 1083.1490478515625
Iteration 1513 logL: 1159.645751953125
Iteration 1514 logL: 1147.1195068359375
Iteration 1515 logL: 1120.597900390625
Iteration 1516 logL: 1076.9443359375
Iteration 1517 logL: 1083.916015625
Iteration 1518 logL: 1091.83251953125
Iteration 1519 logL: 1127.909423828125
Iteration 1520 logL: 1159.16845703125
Iteration 1521 logL: 1086.5340576171875
Iteration 1522 logL: 1053.0126953125
Iteration 1523 logL: 1076.68408203125
Iteration 1524 logL: 1088.9554443359375
Iteration 1525 logL: 1085.61181640625
Iteration 1526 logL: 1047.7569580078125
Iteration 1527 logL: 1040.089111328125
Iteration 1528 logL: 1115.95263671875
Iteration 1529 logL: 1148.160400390625
Iteration 1530 logL: 1074.2188720703125
Iteration 1531 logL: 1072.8055419921875
```

```
Iteration 1532 logL: 1133.7813720703125
Iteration 1533 logL: 1111.05712890625
Iteration 1534 logL: 1076.81298828125
Iteration 1535 logL: 1117.424560546875
Iteration 1536 logL: 1121.345458984375
Iteration 1537 logL: 1005.15673828125
Iteration 1538 logL: 1060.812255859375
Iteration 1539 logL: 1131.9388427734375
Iteration 1540 logL: 1105.917236328125
Iteration 1541 logL: 1047.8763427734375
Iteration 1542 logL: 1016.90771484375
Iteration 1543 logL: 1091.851806640625
Iteration 1544 logL: 1129.99853515625
Iteration 1545 logL: 1114.532958984375
Iteration 1546 logL: 1117.90087890625
Iteration 1547 logL: 1134.5184326171875
Iteration 1548 logL: 1119.154052734375
Iteration 1549 logL: 1082.0614013671875
Iteration 1550 logL: 1131.283203125
Iteration 1551 logL: 1160.9154052734375
Iteration 1552 logL: 1152.974365234375
Iteration 1553 logL: 1143.1324462890625
Iteration 1554 logL: 1146.483642578125
Iteration 1555 logL: 1146.3553466796875
Iteration 1556 logL: 1157.802978515625
Iteration 1557 logL: 1145.6148681640625
Iteration 1558 logL: 1150.203857421875
Iteration 1559 logL: 1158.2666015625
Iteration 1560 logL: 1152.2862548828125
Iteration 1561 logL: 1175.57568359375
Iteration 1562 logL: 1168.501708984375
Iteration 1563 logL: 1170.6280517578125
Iteration 1564 logL: 1142.899169921875
Iteration 1565 logL: 1165.309814453125
Iteration 1566 logL: 1151.5889892578125
Iteration 1567 logL: 1168.0517578125
Iteration 1568 logL: 1137.4246826171875
Iteration 1569 logL: 1146.2012939453125
Iteration 1570 logL: 1152.290283203125
Iteration 1571 logL: 1122.83447265625
Iteration 1572 logL: 1111.770751953125
Iteration 1573 logL: 1129.753173828125
Iteration 1574 logL: 1140.9453125
Iteration 1575 logL: 1104.6373291015625
Iteration 1576 logL: 1144.629638671875
Iteration 1577 logL: 1156.2913818359375
Iteration 1578 logL: 1128.4259033203125
Iteration 1579 logL: 1120.397705078125
Iteration 1580 logL: 1173.950439453125
Iteration 1581 logL: 1151.0477294921875
Iteration 1582 logL: 1139.95068359375
Iteration 1583 logL: 1128.950927734375
Iteration 1584 logL: 1157.8828125
Iteration 1585 logL: 1137.711181640625
Iteration 1586 logL: 1125.2657470703125
Iteration 1587 logL: 1102.0804443359375
Iteration 1588 logL: 1137.258056640625
Iteration 1589 logL: 1147.3424072265625
Iteration 1590 logL: 1119.6539306640625
```

```
Iteration 1591 logL: 1052.132080078125
Iteration 1592 logL: 901.9892578125
Iteration 1593 logL: 889.6141357421875
Iteration 1594 logL: 1029.9930419921875
Iteration 1595 logL: 1126.2838134765625
Iteration 1596 logL: 1076.828857421875
Iteration 1597 logL: 1012.1829833984375
Iteration 1598 logL: 1030.5869140625
Iteration 1599 logL: 1143.045166015625
Iteration 1600 logL: 1081.0469970703125
Iteration 1601 logL: 1033.7611083984375
Iteration 1602 logL: 1137.769775390625
Iteration 1603 logL: 1066.763671875
Iteration 1604 logL: 1072.4102783203125
Iteration 1605 logL: 1111.506591796875
Iteration 1606 logL: 1089.295654296875
Iteration 1607 logL: 1054.1484375
Iteration 1608 logL: 1058.26708984375
Iteration 1609 logL: 1140.2510986328125
Iteration 1610 logL: 1138.364013671875
Iteration 1611 logL: 1042.8646240234375
Iteration 1612 logL: 1091.0401611328125
Iteration 1613 logL: 1146.78515625
Iteration 1614 logL: 1052.33984375
Iteration 1615 logL: 1001.0062255859375
Iteration 1616 logL: 1128.2572021484375
Iteration 1617 logL: 1142.9827880859375
Iteration 1618 logL: 1100.8253173828125
Iteration 1619 logL: 1162.267822265625
Iteration 1620 logL: 1112.606201171875
Iteration 1621 logL: 1084.6873779296875
Iteration 1622 logL: 1133.799560546875
Iteration 1623 logL: 1153.851806640625
Iteration 1624 logL: 1093.0877685546875
Iteration 1625 logL: 1091.386474609375
Iteration 1626 logL: 1140.1612548828125
Iteration 1627 logL: 1116.431640625
Iteration 1628 logL: 1091.160400390625
Iteration 1629 logL: 1176.3953857421875
Iteration 1630 logL: 1154.5194091796875
Iteration 1631 logL: 1121.717529296875
Iteration 1632 logL: 1152.7113037109375
Iteration 1633 logL: 1125.14697265625
Iteration 1634 logL: 1125.85888671875
Iteration 1635 logL: 1157.7203369140625
Iteration 1636 logL: 1159.6851806640625
Iteration 1637 logL: 1162.5452880859375
Iteration 1638 logL: 1152.07470703125
Iteration 1639 logL: 1163.7086181640625
Iteration 1640 logL: 1173.7105712890625
Iteration 1641 logL: 1141.83642578125
Iteration 1642 logL: 1168.4686279296875
Iteration 1643 logL: 1187.3824462890625
Iteration 1644 logL: 1135.5616455078125
Iteration 1645 logL: 1131.251220703125
Iteration 1646 logL: 1152.1446533203125
Iteration 1647 logL: 1131.80517578125
Iteration 1648 logL: 1156.0887451171875
Iteration 1649 logL: 1146.70947265625
```

```
Iteration 1650 logL: 1151.3399658203125
Iteration 1651 logL: 1163.189453125
Iteration 1652 logL: 1141.3582763671875
Iteration 1653 logL: 1143.53564453125
Iteration 1654 logL: 1180.776123046875
Iteration 1655 logL: 1113.526611328125
Iteration 1656 logL: 1149.565185546875
Iteration 1657 logL: 1174.4774169921875
Iteration 1658 logL: 1145.5340576171875
Iteration 1659 logL: 1136.8470458984375
Iteration 1660 logL: 1188.296142578125
Iteration 1661 logL: 1171.7259521484375
Iteration 1662 logL: 1183.34716796875
Iteration 1663 logL: 1143.724609375
Iteration 1664 logL: 1121.6688232421875
Iteration 1665 logL: 1183.287353515625
Iteration 1666 logL: 1182.9942626953125
Iteration 1667 logL: 1135.4873046875
Iteration 1668 logL: 1147.2803955078125
Iteration 1669 logL: 1146.2919921875
Iteration 1670 logL: 1171.281005859375
Iteration 1671 logL: 1183.0555419921875
Iteration 1672 logL: 1159.0999755859375
Iteration 1673 logL: 1089.140869140625
Iteration 1674 logL: 1125.0054931640625
Iteration 1675 logL: 1139.187255859375
Iteration 1676 logL: 1141.2471923828125
Iteration 1677 logL: 1148.109619140625
Iteration 1678 logL: 1157.97705078125
Iteration 1679 logL: 1172.8035888671875
Iteration 1680 logL: 1167.236083984375
Iteration 1681 logL: 1115.010498046875
Iteration 1682 logL: 1060.484619140625
Iteration 1683 logL: 1076.152587890625
Iteration 1684 logL: 1167.9429931640625
Iteration 1685 logL: 1190.0489501953125
Iteration 1686 logL: 1066.48095703125
Iteration 1687 logL: 1076.6883544921875
Iteration 1688 logL: 1161.2801513671875
Iteration 1689 logL: 1153.4073486328125
Iteration 1690 logL: 1151.1829833984375
Iteration 1691 logL: 1147.67041015625
Iteration 1692 logL: 1163.494140625
Iteration 1693 logL: 1133.4549560546875
Iteration 1694 logL: 1109.160888671875
Iteration 1695 logL: 1081.82958984375
Iteration 1696 logL: 1086.3121337890625
Iteration 1697 logL: 1144.2139892578125
Iteration 1698 logL: 1078.1351318359375
Iteration 1699 logL: 1008.3033447265625
Iteration 1700 logL: 1085.6053466796875
Iteration 1701 logL: 1195.57958984375
Iteration 1702 logL: 1052.801513671875
Iteration 1703 logL: 938.021484375
Iteration 1704 logL: 1124.0869140625
Iteration 1705 logL: 1134.34814453125
Iteration 1706 logL: 999.1136474609375
Iteration 1707 logL: 997.519287109375
Iteration 1708 logL: 1091.32421875
```

```
Iteration 1709 logL: 1090.6494140625
Iteration 1710 logL: 1103.7757568359375
Iteration 1711 logL: 1104.9930419921875
Iteration 1712 logL: 1052.1116943359375
Iteration 1713 logL: 1181.619384765625
Iteration 1714 logL: 1139.8482666015625
Iteration 1715 logL: 1089.2135009765625
Iteration 1716 logL: 1156.0135498046875
Iteration 1717 logL: 1163.00634765625
Iteration 1718 logL: 1075.65087890625
Iteration 1719 logL: 1140.47119140625
Iteration 1720 logL: 1180.4326171875
Iteration 1721 logL: 1154.65087890625
Iteration 1722 logL: 1165.97216796875
Iteration 1723 logL: 1185.676513671875
Iteration 1724 logL: 1161.33935546875
Iteration 1725 logL: 1158.6217041015625
Iteration 1726 logL: 1189.35205078125
Iteration 1727 logL: 1136.858154296875
Iteration 1728 logL: 1150.0458984375
Iteration 1729 logL: 1180.3701171875
Iteration 1730 logL: 1157.0469970703125
Iteration 1731 logL: 1165.01025390625
Iteration 1732 logL: 1144.8123779296875
Iteration 1733 logL: 1155.8565673828125
Iteration 1734 logL: 1141.1715087890625
Iteration 1735 logL: 1088.83837890625
Iteration 1736 logL: 1172.6944580078125
Iteration 1737 logL: 1180.142333984375
Iteration 1738 logL: 1113.2823486328125
Iteration 1739 logL: 1139.8388671875
Iteration 1740 logL: 1173.06201171875
Iteration 1741 logL: 1120.55126953125
Iteration 1742 logL: 1149.968505859375
Iteration 1743 logL: 1187.1422119140625
Iteration 1744 logL: 1172.948974609375
Iteration 1745 logL: 1146.9542236328125
Iteration 1746 logL: 1195.833251953125
Iteration 1747 logL: 1197.7987060546875
Iteration 1748 logL: 1190.53515625
Iteration 1749 logL: 1200.8770751953125
Iteration 1750 logL: 1202.64404296875
Iteration 1751 logL: 1176.536376953125
Iteration 1752 logL: 1204.62353515625
Iteration 1753 logL: 1171.98046875
Iteration 1754 logL: 1138.4837646484375
Iteration 1755 logL: 1168.888671875
Iteration 1756 logL: 1123.74951171875
Iteration 1757 logL: 1099.937744140625
Iteration 1758 logL: 1099.6246337890625
Iteration 1759 logL: 1095.86669921875
Iteration 1760 logL: 1129.895263671875
Iteration 1761 logL: 1061.75341796875
Iteration 1762 logL: 981.8104248046875
Iteration 1763 logL: 1072.3914794921875
Iteration 1764 logL: 1060.898193359375
Iteration 1765 logL: 1064.474365234375
Iteration 1766 logL: 1071.06640625
Iteration 1767 logL: 1029.953369140625
```

```
Iteration 1768 logL: 1157.8651123046875
Iteration 1769 logL: 1072.356201171875
Iteration 1770 logL: 1094.0537109375
Iteration 1771 logL: 1134.451416015625
Iteration 1772 logL: 1059.017822265625
Iteration 1773 logL: 1029.544189453125
Iteration 1774 logL: 1103.8106689453125
Iteration 1775 logL: 1160.5919189453125
Iteration 1776 logL: 1131.671142578125
Iteration 1777 logL: 1128.98046875
Iteration 1778 logL: 1189.90185546875
Iteration 1779 logL: 1136.0472412109375
Iteration 1780 logL: 1103.69775390625
Iteration 1781 logL: 1176.66650390625
Iteration 1782 logL: 1156.9439697265625
Iteration 1783 logL: 1018.382568359375
Iteration 1784 logL: 1121.1993408203125
Iteration 1785 logL: 1174.5164794921875
Iteration 1786 logL: 1138.939697265625
Iteration 1787 logL: 1201.796630859375
Iteration 1788 logL: 1161.7603759765625
Iteration 1789 logL: 1155.3951416015625
Iteration 1790 logL: 1175.95263671875
Iteration 1791 logL: 1190.130126953125
Iteration 1792 logL: 1173.4368896484375
Iteration 1793 logL: 1192.296875
Iteration 1794 logL: 1189.837646484375
Iteration 1795 logL: 1180.27734375
Iteration 1796 logL: 1200.889892578125
Iteration 1797 logL: 1175.4337158203125
Iteration 1798 logL: 1159.1953125
Iteration 1799 logL: 1195.88671875
Iteration 1800 logL: 1186.492919921875
Iteration 1801 logL: 1154.389404296875
Iteration 1802 logL: 1210.261474609375
Iteration 1803 logL: 1188.316162109375
Iteration 1804 logL: 1147.332275390625
Iteration 1805 logL: 1172.823974609375
Iteration 1806 logL: 1190.45556640625
Iteration 1807 logL: 1187.948974609375
Iteration 1808 logL: 1190.2025146484375
Iteration 1809 logL: 1163.7408447265625
Iteration 1810 logL: 1209.648681640625
Iteration 1811 logL: 1209.9573974609375
Iteration 1812 logL: 1181.302734375
Iteration 1813 logL: 1199.8917236328125
Iteration 1814 logL: 1229.9256591796875
Iteration 1815 logL: 1146.1197509765625
Iteration 1816 logL: 1150.55810546875
Iteration 1817 logL: 1197.42919921875
Iteration 1818 logL: 1208.4410400390625
Iteration 1819 logL: 1229.9888916015625
Iteration 1820 logL: 1191.4019775390625
Iteration 1821 logL: 1202.1630859375
Iteration 1822 logL: 1183.2589111328125
Iteration 1823 logL: 1184.36865234375
Iteration 1824 logL: 1179.5299072265625
Iteration 1825 logL: 1195.7923583984375
Iteration 1826 logL: 1207.087890625
```

```
Iteration 1827 logL: 1137.3955078125
Iteration 1828 logL: 1137.027587890625
Iteration 1829 logL: 1150.9666748046875
Iteration 1830 logL: 1153.437255859375
Iteration 1831 logL: 1195.476318359375
Iteration 1832 logL: 1201.533935546875
Iteration 1833 logL: 1209.5994873046875
Iteration 1834 logL: 1133.618896484375
Iteration 1835 logL: 1119.32373046875
Iteration 1836 logL: 1163.4991455078125
Iteration 1837 logL: 1176.7540283203125
Iteration 1838 logL: 1182.48046875
Iteration 1839 logL: 1180.406982421875
Iteration 1840 logL: 1139.6094970703125
Iteration 1841 logL: 1161.464599609375
Iteration 1842 logL: 1150.23486328125
Iteration 1843 logL: 1201.280517578125
Iteration 1844 logL: 1171.5853271484375
Iteration 1845 logL: 1145.40234375
Iteration 1846 logL: 1153.2955322265625
Iteration 1847 logL: 1182.4085693359375
Iteration 1848 logL: 1182.6571044921875
Iteration 1849 logL: 1121.10498046875
Iteration 1850 logL: 1168.7552490234375
Iteration 1851 logL: 1215.7532958984375
Iteration 1852 logL: 1168.606201171875
Iteration 1853 logL: 1156.6822509765625
Iteration 1854 logL: 1185.0814208984375
Iteration 1855 logL: 1208.6463623046875
Iteration 1856 logL: 1177.5478515625
Iteration 1857 logL: 1141.1156005859375
Iteration 1858 logL: 1191.260498046875
Iteration 1859 logL: 1197.61767578125
Iteration 1860 logL: 1227.127197265625
Iteration 1861 logL: 1163.735107421875
Iteration 1862 logL: 1154.60986328125
Iteration 1863 logL: 1187.0550537109375
Iteration 1864 logL: 1213.536865234375
Iteration 1865 logL: 1185.064453125
Iteration 1866 logL: 1157.4537353515625
Iteration 1867 logL: 1200.761962890625
Iteration 1868 logL: 1222.02783203125
Iteration 1869 logL: 1147.2158203125
Iteration 1870 logL: 1144.282958984375
Iteration 1871 logL: 1199.695068359375
Iteration 1872 logL: 1214.810791015625
Iteration 1873 logL: 1194.4130859375
Iteration 1874 logL: 1198.478515625
Iteration 1875 logL: 1210.9346923828125
Iteration 1876 logL: 1211.574462890625
Iteration 1877 logL: 1205.450439453125
Iteration 1878 logL: 1172.6024169921875
Iteration 1879 logL: 1174.087646484375
Iteration 1880 logL: 1203.639892578125
Iteration 1881 logL: 1190.721435546875
Iteration 1882 logL: 1192.437744140625
Iteration 1883 logL: 1220.7801513671875
Iteration 1884 logL: 1226.697509765625
Iteration 1885 logL: 1211.2330322265625
```

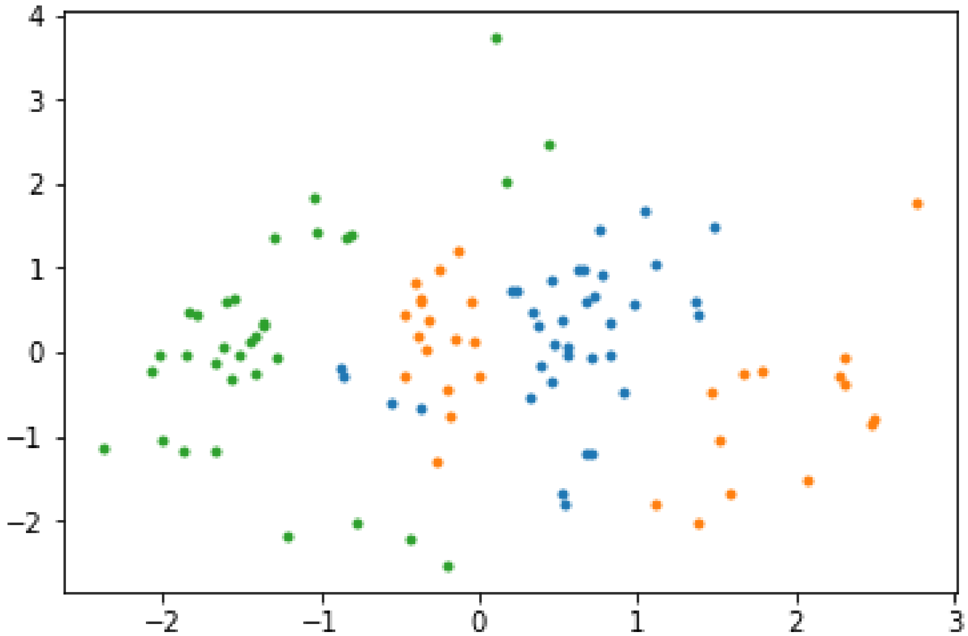
```
Iteration 1886 logL: 1160.687255859375
Iteration 1887 logL: 1158.32666015625
Iteration 1888 logL: 1117.36279296875
Iteration 1889 logL: 1082.3958740234375
Iteration 1890 logL: 1176.63623046875
Iteration 1891 logL: 1232.69580078125
Iteration 1892 logL: 1152.8900146484375
Iteration 1893 logL: 1107.21142578125
Iteration 1894 logL: 1147.4429931640625
Iteration 1895 logL: 1213.348388671875
Iteration 1896 logL: 1174.9425048828125
Iteration 1897 logL: 1107.7596435546875
Iteration 1898 logL: 1107.3375244140625
Iteration 1899 logL: 1192.389404296875
Iteration 1900 logL: 1214.0885009765625
Iteration 1901 logL: 1117.0269775390625
Iteration 1902 logL: 1143.033447265625
Iteration 1903 logL: 1161.20361328125
Iteration 1904 logL: 1222.047119140625
Iteration 1905 logL: 1183.93896484375
Iteration 1906 logL: 1175.2034912109375
Iteration 1907 logL: 1191.1968994140625
Iteration 1908 logL: 1223.955810546875
Iteration 1909 logL: 1210.2122802734375
Iteration 1910 logL: 1181.78759765625
Iteration 1911 logL: 1210.9541015625
Iteration 1912 logL: 1204.458984375
Iteration 1913 logL: 1196.3955078125
Iteration 1914 logL: 1206.798095703125
Iteration 1915 logL: 1198.698974609375
Iteration 1916 logL: 1188.166259765625
Iteration 1917 logL: 1203.7275390625
Iteration 1918 logL: 1186.60546875
Iteration 1919 logL: 1183.64013671875
Iteration 1920 logL: 1230.890380859375
Iteration 1921 logL: 1194.9090576171875
Iteration 1922 logL: 1123.248779296875
Iteration 1923 logL: 1112.940673828125
Iteration 1924 logL: 1139.887939453125
Iteration 1925 logL: 1239.1982421875
Iteration 1926 logL: 1157.0389404296875
Iteration 1927 logL: 1124.8778076171875
Iteration 1928 logL: 1168.7947998046875
Iteration 1929 logL: 1134.8958740234375
Iteration 1930 logL: 1171.053955078125
Iteration 1931 logL: 1164.15185546875
Iteration 1932 logL: 1136.9044189453125
Iteration 1933 logL: 1129.55078125
Iteration 1934 logL: 1136.8323974609375
Iteration 1935 logL: 1210.345703125
Iteration 1936 logL: 1214.7027587890625
Iteration 1937 logL: 1189.9776611328125
Iteration 1938 logL: 1247.0433349609375
Iteration 1939 logL: 1231.4537353515625
Iteration 1940 logL: 1184.2755126953125
Iteration 1941 logL: 1211.376708984375
Iteration 1942 logL: 1223.527587890625
Iteration 1943 logL: 1248.0943603515625
Iteration 1944 logL: 1256.9500732421875
```

```
Iteration 1945 logL: 1241.0760498046875
Iteration 1946 logL: 1246.47509765625
Iteration 1947 logL: 1227.07080078125
Iteration 1948 logL: 1244.85693359375
Iteration 1949 logL: 1223.26611328125
Iteration 1950 logL: 1218.07666015625
Iteration 1951 logL: 1259.81201171875
Iteration 1952 logL: 1214.4500732421875
Iteration 1953 logL: 1218.712646484375
Iteration 1954 logL: 1271.29638671875
Iteration 1955 logL: 1250.2908935546875
Iteration 1956 logL: 1225.498291015625
Iteration 1957 logL: 1218.0775146484375
Iteration 1958 logL: 1243.34521484375
Iteration 1959 logL: 1228.5477294921875
Iteration 1960 logL: 1213.529052734375
Iteration 1961 logL: 1214.8626708984375
Iteration 1962 logL: 1231.824462890625
Iteration 1963 logL: 1230.359619140625
Iteration 1964 logL: 1250.611572265625
Iteration 1965 logL: 1241.3177490234375
Iteration 1966 logL: 1239.142822265625
Iteration 1967 logL: 1231.14208984375
Iteration 1968 logL: 1259.7193603515625
Iteration 1969 logL: 1221.607421875
Iteration 1970 logL: 1226.014404296875
Iteration 1971 logL: 1216.6728515625
Iteration 1972 logL: 1205.4998779296875
Iteration 1973 logL: 1214.89208984375
Iteration 1974 logL: 1222.856689453125
Iteration 1975 logL: 1237.81689453125
Iteration 1976 logL: 1208.667236328125
Iteration 1977 logL: 1146.57763671875
Iteration 1978 logL: 1131.6876220703125
Iteration 1979 logL: 1097.2244873046875
Iteration 1980 logL: 1074.828369140625
Iteration 1981 logL: 1098.2362060546875
Iteration 1982 logL: 1106.648681640625
Iteration 1983 logL: 1064.2252197265625
Iteration 1984 logL: 1102.831787109375
Iteration 1985 logL: 1179.8662109375
Iteration 1986 logL: 1138.1878662109375
Iteration 1987 logL: 1111.7362060546875
Iteration 1988 logL: 1141.1280517578125
Iteration 1989 logL: 1190.904296875
Iteration 1990 logL: 1222.82470703125
Iteration 1991 logL: 1200.0384521484375
Iteration 1992 logL: 1195.4183349609375
Iteration 1993 logL: 1223.42919921875
Iteration 1994 logL: 1202.7012939453125
Iteration 1995 logL: 1181.566650390625
Iteration 1996 logL: 1211.32568359375
Iteration 1997 logL: 1247.471435546875
Iteration 1998 logL: 1243.31005859375
Iteration 1999 logL: 1203.407470703125
Iteration 2000 logL: 1221.434326171875
```

#### Plot the training data in the latent space

```
In [13]: q_x_mean = q.encoder.block(mx.nd.array(Y)).asnumpy()
```

```
In [14]: for i in range(3):
             plot(q_x_mean[label==i,0], q_x_mean[label==i,1], '.')
```




## 4.3 Developer Tutorials

### 4.3.1 Writing a new Distribution

To write and and use a new Distribution class in MXFusion, fill out the Distribution interface and either the Univariate or Multivariate interface, depending on the type of distribution you are creating.

There are 4 primary methods to fill out for a Distribution in MXFusion: * `__init__` - This is the constructor for the Distribution. It takes in any parameters the distribution needs. It also defines names for the input variable[s] that the distribution takes and the output variable[s] it produces. * `log_pdf` - This method returns the logarithm of probabilistic density function for the distribution. This is called during Inference time as necessary to perform the Inference algorithm. * `draw_samples` - This method returns drawn samples from the distribution. This is called during Inference time as necessary to perform the Inference algorithm. * `define_variable` - This is used to generate random variables drawn from the Distribution used during model definition.

`log_pdf` and `draw_samples` are implemented using MXNet functions to compute on the input variables, which at Inference time are MXNet arrays or MXNet symbolic variables.

This notebook will take the Normal distribution as a reference.

#### File Structure

Code for distributions lives in the mxfusion/components/distributions directory.

If you're implementing the *FancyNew* distribution then you should create a file called mxfusion/components/distributions/fancy_new.py for the class to live in.

### Interface Implementation

Since this example is for a Univariate Normal distribution, our class extends the UnivatiateDistribution class.

The Normal distribution's constructor takes in objects for its `mean` and `variance`, specifications for data type and context, and a random number generator if not the default.

In addition, a distribution can take in additional parameters used for calculating that aren't inputs. We refer to these additional parameters as the Distribution's `attributes`. The difference between an input and an attribute is primarily that inputs are dynamic at inference time, while attributes are static throughout a given inference run.

In this case, `minibatch_ratio` is a static attribute, as it doesn't change for a given minibatch size during inference.

The mean and variance can be either Variables or MXNet arrays if they are constants.

As mentioned above, you define names for the input and output variable[s] for the distribution here. These names are used when printing and generally inspecting the model, so give meaningful names. We prefer names like `mean` and `variance` to ones like `location` and `scale` or greek letters like `mew` and `sigma`.

```
In [ ]: class Normal(UnivariateDistribution):
            """
            The one-dimensional normal distribution. The normal distribution can be defined over a sc

            :param mean: Mean of the normal distribution.
            :type mean: Variable
            :param variance: Variance of the normal distribution.
            :type variance: Variable
            :param rand_gen: the random generator (default: MXNetRandomGenerator)
            :type rand_gen: RandomGenerator
            :param dtype: the data type for float point numbers
            :type dtype: numpy.float32 or numpy.float64
            :param ctx: the mxnet context (default: None/current context)
            :type ctx: None or mxnet.cpu or mxnet.gpu
            """
            def __init__(self, mean, variance, rand_gen=None, minibatch_ratio=1.,
                         dtype=None, ctx=None):
                self.minibatch_ratio = minibatch_ratio
                if not isinstance(mean, Variable):
                    mean = Variable(value=mean)
                if not isinstance(variance, Variable):
                    variance = Variable(value=variance)

                inputs = [('mean', mean), ('variance', variance)]
                input_names = ['mean', 'variance']
                output_names = ['random_variable']
                super(Normal, self).__init__(inputs=inputs, outputs=None,
                                             input_names=input_names,
                                             output_names=output_names,
                                             rand_gen=rand_gen, dtype=dtype, ctx=ctx)
```

If your distribution's `__init__` function only takes in parameters that get passed onto its super constructor, you don't need to implement `replicate_self`. If it does take additional parameters (as the Normal distribution does for minibatch_ratio), those parameters need to be copied over to the replicant Distribution before returning, as below.

```
In [ ]:     def replicate_self(self, attribute_map=None):
                """
                Replicates this Factor, using new inputs, outputs, and a new uuid.
```

```
        Used during model replication to functionally replicate a factor into a new graph.
        :param inputs: new input variables of the factor
        :type inputs: a dict of {'name' : Variable} or None
        :param outputs: new output variables of the factor.
        :type outputs: a dict of {'name' : Variable} or None
        """
        replicant = super(Normal, self).replicate_self(attribute_map)
        replicant.minibatch_ratio = self.minibatch_ratio
        return replicant
```

`log_pdf` and `draw_samples` are relatively straightforward for implementation. These are the meaningful parts of the Distribution that you're implementing, putting the math into code using MXNet operators for the compute.

If it's a distribution that isn't well documented on Wikipedia, please add a link to a paper or other resource that explains what it's doing and why.

```
In [ ]:     def log_pdf(self, mean, variance, random_variable, F=None):
        """
        Computes the logarithm of the probability density function (PDF) of the normal distr

        :param mean: the mean of the normal distribution
        :type mean: MXNet NDArray or MXNet Symbol
        :param variance: the variance of the normal distributions
        :type variance: MXNet NDArray or MXNet Symbol
        :param random_variable: the random variable of the normal distribution
        :type random_variable: MXNet NDArray or MXNet Symbol
        :param F: the MXNet computation mode (mxnet.symbol or mxnet.ndarray)
        :returns: log pdf of the distribution
        :rtypes: MXNet NDArray or MXNet Symbol
        """
        F = get_default_MXNet_mode() if F is None else F
        logvar = np.log(2 * np.pi) / -2 + F.log(variance) / -2
        logL = F.broadcast_add(logvar, F.broadcast_div(F.square(
            F.broadcast_minus(random_variable, mean)), -2 * variance))
        return logL
```

```
In [ ]:     def draw_samples(self, mean, variance, rv_shape, num_samples=1, F=None):
        """
        Draw samples from the normal distribution.

        :param mean: the mean of the normal distribution
        :type mean: MXNet NDArray or MXNet Symbol
        :param variance: the variance of the normal distributions
        :type variance: MXNet NDArray or MXNet Symbol
        :param rv_shape: the shape of each sample
        :type rv_shape: tuple
        :param num_samples: the number of drawn samples (default: one)
        :int num_samples: int
        :param F: the MXNet computation mode (mxnet.symbol or mxnet.ndarray)
        :returns: a set samples of the normal distribution
        :rtypes: MXNet NDArray or MXNet Symbol
        """
        F = get_default_MXNet_mode() if F is None else F
        out_shape = (num_samples,) + rv_shape
        return F.broadcast_add(F.broadcast_mul(self._rand_gen.sample_normal(
            shape=out_shape, dtype=self.dtype, ctx=self.ctx),
            F.sqrt(variance)), mean)
```

`define_variable` is just a helper function for end users. All it does is take in parameters for the distribution, create a distribution based on those parameters, then return the output variables of that distribution.

In [ ]:
```
@staticmethod
def define_variable(mean=0., variance=1., shape=None, rand_gen=None,
                    minibatch_ratio=1., dtype=None, ctx=None):
    """
    Creates and returns a random variable drawn from a normal distribution.

    :param mean: Mean of the distribution.
    :param variance: Variance of the distribution.
    :param shape: the shape of the random variable(s)
    :type shape: tuple or [tuple]
    :param rand_gen: the random generator (default: MXNetRandomGenerator)
    :type rand_gen: RandomGenerator
    :param dtype: the data type for float point numbers
    :type dtype: numpy.float32 or numpy.float64
    :param ctx: the mxnet context (default: None/current context)
    :type ctx: None or mxnet.cpu or mxnet.gpu
    :returns: the random variables drawn from the normal distribution.
    :rtypes: Variable
    """
    normal = Normal(mean=mean, variance=variance, rand_gen=rand_gen,
                    dtype=dtype, ctx=ctx)
    normal._generate_outputs(shape=shape)
    return normal.random_variable
```

### Using your new distribution

At this point, you should be ready to start testing your new Distribution's functionality by importing it like any other MXFusion component.

### Testing

Before submitting your new code as a pull request, please write unit tests that verify it works as expected. This should include numerical checks against edge cases. See the existing test cases for the Normal or Categorical distributions for example tests.

# CHAPTER 5

# Indices and tables

- genindex
- modindex
- search