python: how to plot one line in different colors

Asked 8 years, 5 months ago Active 7 months ago Viewed 38k times

16

12

I have two list as below:

```
latt=
[42.0,41.978567980875397,41.96622693388357,41.963791391892457,...,41.972407378075879]
lont=
[-66.706920989908909,-66.703116557977069,-66.707351643324543,...-66.718218142021925]
```

now I want to plot this as a line, separate each 10 of those 'latt' and 'lont' records as a period and give it a unique color. what should I do?

python matplotlib Edit tags

Share Edit Follow Close Flag

asked Jun 21 '13 at 17:04
wuwucat
2,165 8 22 25

Look at `scatter` matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.scatter – tacaswell Jun 21 '13 at 17:08

6 Answers

Active Oldest Votes

0

I have been searching for a short solution how to use pyplots line plot to show a time series coloured by a label feature **without using scatter** due to the amount of data points.

I came up with the following workaround:

```
plt.plot(np.where(df["label"]==1, df["myvalue"], None), color="red", label="1")
plt.plot(np.where(df["label"]==0, df["myvalue"], None), color="blue", label="0")
plt.legend()
```

The drawback is you are creating two different line plots so the connection between the different classes is not shown. For my purposes it is not a big deal. It may help someone.

Share Edit Follow Flag

answered May 4 at 8:56
Andre S.
414 4 12

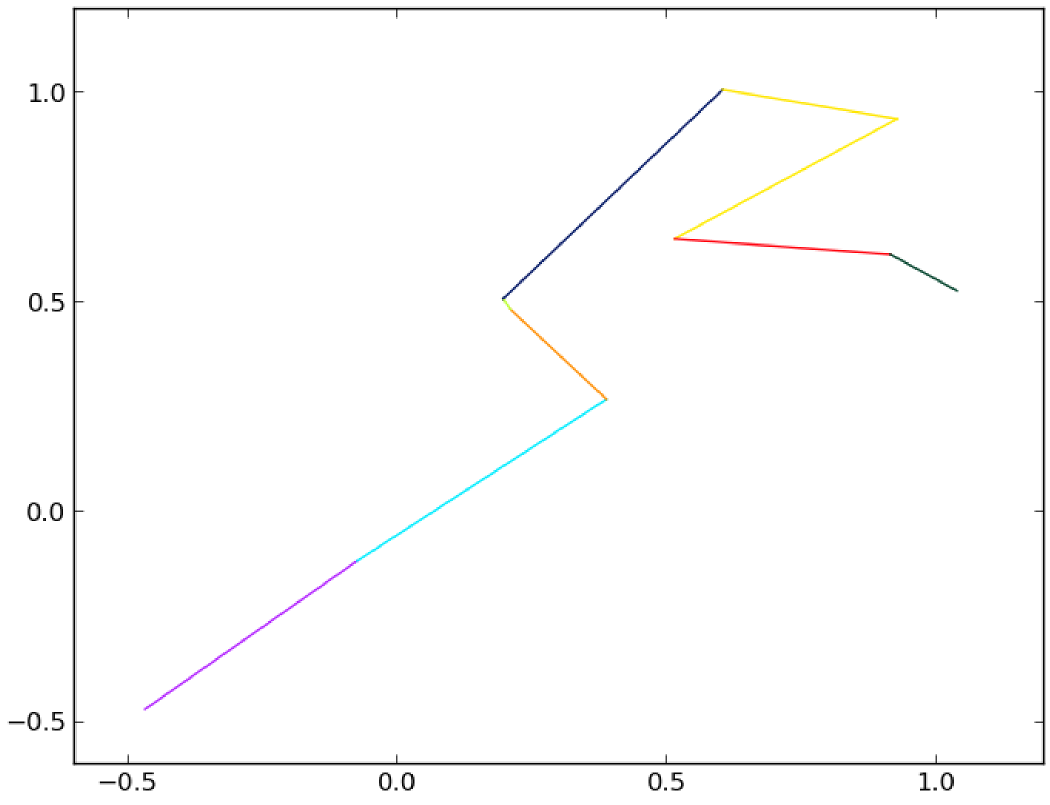
There are several different ways to do this. The "best" approach will depend mostly on how

38

many line segments you want to plot.

If you're just going to be plotting a handful (e.g. 10) line segments, then just do something like:

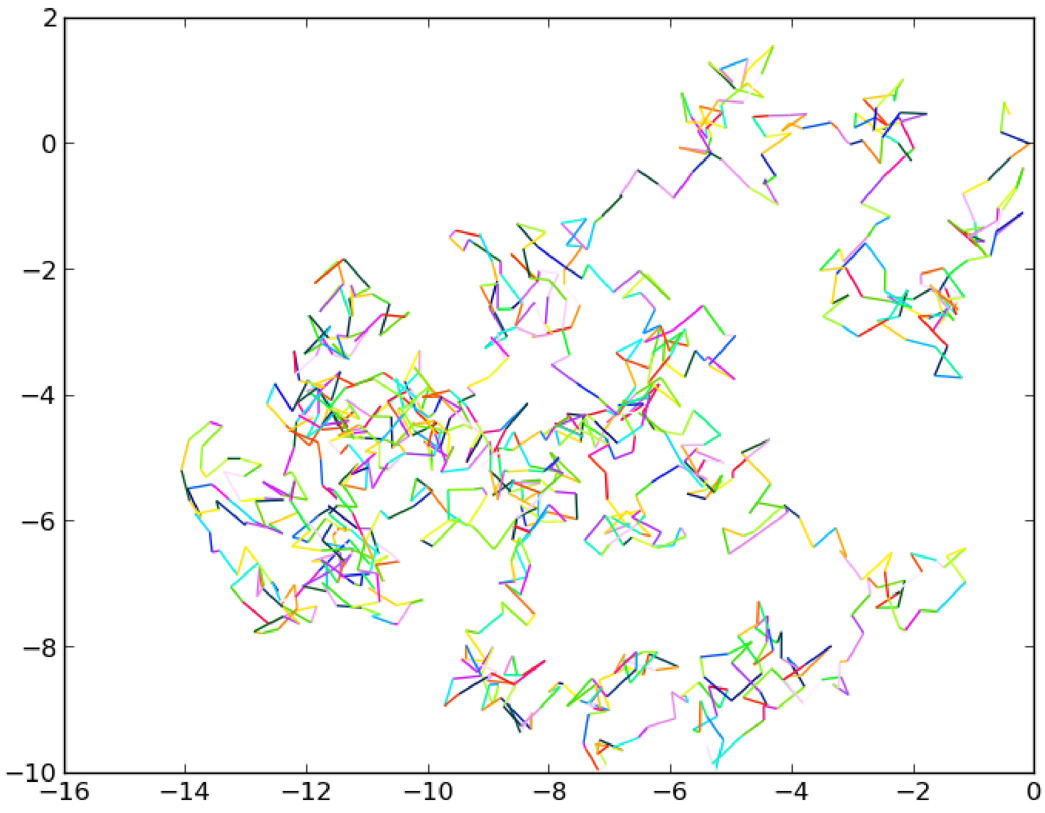
```
import numpy as np
import matplotlib.pyplot as plt

def uniqueish_color():
    """There're better ways to generate unique colors, but this isn't awful."""
    return plt.cm.gist_ncar(np.random.random())

xy = (np.random.random((10, 2)) - 0.5).cumsum(axis=0)

fig, ax = plt.subplots()
for start, stop in zip(xy[:-1], xy[1:]):
    x, y = zip(start, stop)
    ax.plot(x, y, color=uniqueish_color())
plt.show()
```



If you're plotting something with a million line segments, though, this will be terribly slow to draw. In that case, use a `LineCollection`. E.g.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.collections import LineCollection

xy = (np.random.random((1000, 2)) - 0.5).cumsum(axis=0)

# Reshape things so that we have a sequence of:
```

```
# [[(x0,y0),(x1,y1)],[(x0,y0),(x1,y1)],...]
xy = xy.reshape(-1, 1, 2)
segments = np.hstack([xy[:-1], xy[1:]])

fig, ax = plt.subplots()
coll = LineCollection(segments, cmap=plt.cm.gist_ncar)
coll.set_array(np.random.random(xy.shape[0]))

ax.add_collection(coll)
ax.autoscale_view()

plt.show()
```



For both of these cases, we're just drawing random colors from the "gist_ncar" coloramp. Have a look at the colormaps here (gist_ncar is about 2/3 of the way down): http://matplotlib.org/examples/color/colormaps_reference.html

Share Edit Follow Flag

edited Jan 18 '17 at 18:43
tacaswell
77.2k 19 196 186

answered Jun 21 '13 at 17:47
Joe Kington
251k 66 566 450

Ah, I assumed he wanted lines from the "now I want to plot this as a line", but on re-reading, you're probably right. – Joe Kington Jun 21 '13 at 17:56

question. What are you specifying with: `coll.set_array(np.random.random(xy.shape[0]))` documentation is very unclear about this link – J.A.Cado Feb 8 '19 at 8:39

@JoeKington this looks awesome, I have a question, How can I make it limited to two lines, if the value is negative, shows red otherwise green – Volatil3 Oct 20 at 5:49

2

Cribbing the color choice off of @JoeKington,

```
import numpy as np
import matplotlib.pyplot as plt

def uniqueish_color(n):
    """There're better ways to generate unique colors, but this isn't awful."""
    return plt.cm.gist_ncar(np.random.random(n))

plt.scatter(latt, lont, c=uniqueish_color(len(latt)))
```

You can do this with `scatter`.

Share Edit Follow Flag

answered Jun 21 '13 at 18:36
tacaswell
77.2k 19 196 186

This fails with 'Invalid RGBA argument' in Matlplotlib 3.0.0 – Nibor Mar 21 '19 at 9:46

0

This post is hidden. It was deleted 8 years ago by the post author.

You can do something like this to plot in intervals of 10:

```
import matplotlib.pyplot as plt
p = 10
for i in range(len(latt))[::p]:
     plt.scatter( latt[i:i+p], lont[i:i+p], 'o' )
```

Setting colors:

```
import numpy as np

ax = plt.gca()
color_floats = np.linspace(0,1,len(ax.lines))
ax.set_color_cycle( (plt.cm.rainbow(x) for x in color_floats) )
```

`plt.cm` is a module with many colormaps... among them the `rainbow` which goes from `0` to `1` through the colors in the rainbow (`0` closer to `red` and `1` to `violet`).

Share Edit Follow Flag

edited Jun 21 '13 at 17:37

answered Jun 21 '13 at 17:22
Saullo G. P. Castro
51.5k 24 166 228

Comments disabled on deleted / locked posts / reviews

Copied from this example:

5

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.collections import LineCollection
from matplotlib.colors import ListedColormap, BoundaryNorm

x = np.linspace(0, 3 * np.pi, 500)
y = np.sin(x)
z = np.cos(0.5 * (x[:-1] + x[1:]))  # first derivative

# Create a colormap for red, green and blue and a norm to color
# f' < -0.5 red, f' > 0.5 blue, and the rest green
cmap = ListedColormap(['r', 'g', 'b'])
norm = BoundaryNorm([-1, -0.5, 0.5, 1], cmap.N)

# Create a set of line segments so that we can color them individually
# This creates the points as a N x 1 x 2 array so that we can stack points
# together easily to get the segments. The segments array for line collection
# needs to be numlines x points per line x 2 (x and y)
points = np.array([x, y]).T.reshape(-1, 1, 2)
segments = np.concatenate([points[:-1], points[1:]], axis=1)

# Create the line collection object, setting the colormapping parameters.
# Have to set the actual values used for colormapping separately.
lc = LineCollection(segments, cmap=cmap, norm=norm)
lc.set_array(z)
lc.set_linewidth(3)

fig1 = plt.figure()
plt.gca().add_collection(lc)
plt.xlim(x.min(), x.max())
plt.ylim(-1.1, 1.1)

plt.show()
```

Share Edit Follow Flag

edited Jun 21 '13 at 17:26

answered Jun 21 '13 at 17:20
ali_m
65.4k 16 209 281

2

See the answer here to generate the "periods" and then use the matplotlib scatter function as @tcaswell mentioned. Using the plot.hold function you can plot each period, colors will increment automatically.

Share Edit Follow Flag

edited May 23 '17 at 12:10
Community Bot
1 1

answered Jun 21 '13 at 17:15
blazetopher
1,030 8 13