Microsoft: DAT210x Programming with Python for Data Science

Bookmarks

- Start Here
- 1. The Big Picture
- 2. Data And Features
- 3. Exploring Data
- **4. Transforming Data**

Lecture: Transformations

Lecture: PCA
Quiz

Lab: PCA
Lab

Lecture: Isomap
Quiz

Lab: Isomap
Lab

Lecture: Data Cleansing
Quiz

4. Transforming Data > Lab: Isomap > Assignment 5

Bookmark

Lab Assignment 5

Now that you've had your first taste of isomap, let's take your knowledge of it to the next level.

Whatever your high-dimensional samples are, be they images, sound files, or thoughtfully collected attributes, they can all be considered single points in a high dimensional feature-space. Each one of your observations is just a *single* point. Even with a high dimensionality, it's possible that most or all your samples actually lie on a lower dimension surface. Isomap aims to capture that embedding, which is essentially the motion in the underlying, non-linear degrees of freedom.

By testing isomap on a carefully constructed dataset, you will be able to visually confirm its effectiveness, and gain a deeper understanding of how and why each parameter acts the way it does. The ALOI, Amsterdam Library of Object Images, hosts a huge collection of 1000 small objects that were photographed in such a controlled environment, by systematically varying the viewing angle, illumination angle, and illumination color for each object *separately*. To really drive home how well isomap does what it claims, this lab will make use of two image sets taken from the ALOI's collection.

Dive Deeper

▸ 5. Data Modeling

Manifold extraction, and isomap specifically are really good with vision recognition problems, speech problems, and many other real-world tasks, such as identifying similar objects, or objects that have undergone some change. In the case of the 3D rotating object such as the office chair example from earlier, if every pixel is a feature, at the end of the day, the manifold surface is parametrizable by *just* the angle of the chair—a single feature!

1. Start by having a look through the Module4/Datasets/ALOI/ directory. There are two directories filled with 192 x 144 pixel images. Identify their ordering and try to figure out what's changing between the images. They might not be perfectly ordered, but that doesn't matter to isomap.
2. Create a regular Python list object. Then, write a for-loop that iterates over the images in the Module4/Datasets/ALOI/32/ folder, appending each of them to your list. Each .PNG image should first be loaded into a temporary NDArray, just as shown in the Feature Representation reading.

   Optional: Resample your images down by a factor of two if you have a slower computer. You can also convert the image from `0-255` *to* `0.0-1.0` *if you'd like, but that will have no effect on the algorithm's results.*
3. Convert the list to a dataframe and run isomap on it to compute the lower dimensional embedding. Be sure to set `n_components` to 3 so you can visualize your manifold. You can also set the neighborhood size to six.
4. Plot the first two manifold components using a 2D scatter plot, then plot the first three components using a 3D scatter plot. Run your assignment and then answer the questions below.

Lab Questions

(2/2 points)

Please describe the results of your isomap embedding--either the 3D or 2D one, it doesn't matter:

○ It is completely sporadic. It's hard to detect the pattern at this point, since we discarded too many dimensions

● The embedding appears to follow an easily traversable, 3D spline ✓

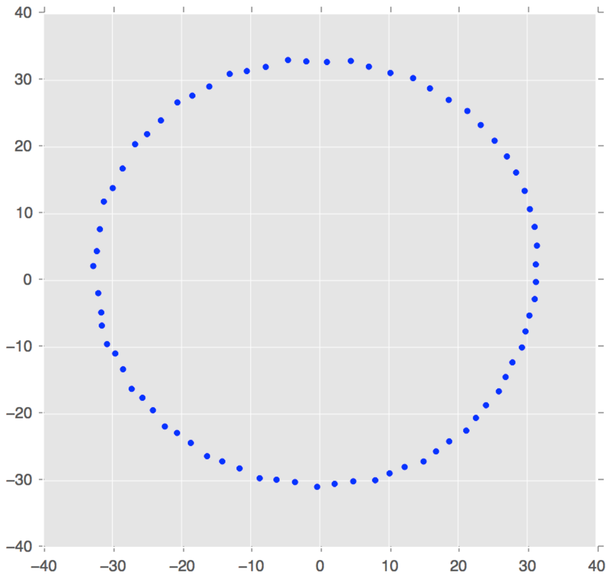
○ It looks like a geometric pattern, but we probably need to increase the neighborhood resolution before a discernible shape emerges

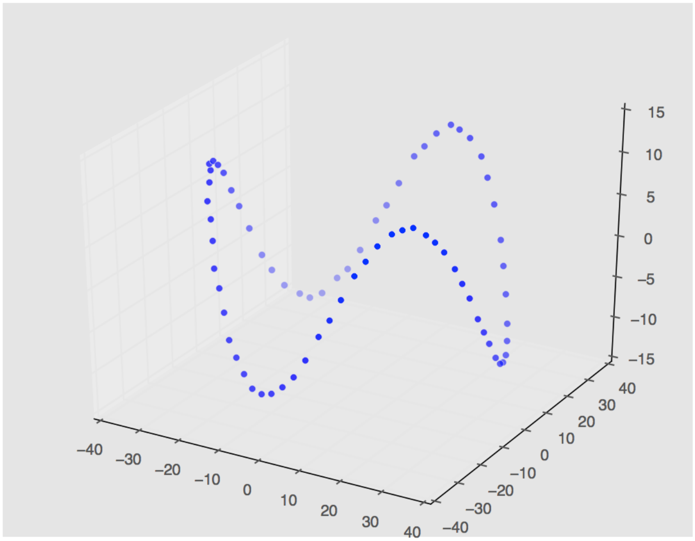
○ Isomap rendered the result as a straight line, as expected, since only a single degree of freedom is altered in the images

EXPLANATION

Encode each image as a single sample. Since SciPy loads images into an NDArray, rather than converting the individual NDArrays to Pandas DataFrames and concatenating DataFrames, it makes more sense to just append all the NDArrays to a list and mass convert the thing in one go.

When rendering the manifold, you should see a smooth curve. In 2D, it should look like a near perfect circle (check your window axes scale, if your X-Axis is stretched). In 3D, it should look like the circle as well, but with the Y-axis values oscillating up and down as if bound to a sine curve:





Try reducing the 'n_neighbors' parameter one value at a time. Keep re-running your assignment until the results look visible different. What is the smallest neighborhood size you can have, while maintaining similar manifold embedding results?

2 ▾ ✓ **Answer:** 2

EXPLANATION

The correct number is two. The dataset was generared carefully enough that there really shouldn't be any cause for noise in the path, unless you introduced it.

You have used 2 of 2 submissions

Almost done! Two more steps to complete this lab:

1. Once you're done answering the first three questions, right before you converted your list to a dataframe, add in additional code which also appends to your list the images in the Module4/Datasets/ALOI/32_i directory.
2. Create a *colors* Python list. Store a `'b'` in it for each element you load from the /32/ directory, and an `'r'` for each element you load from the '32_i' directory. Then pass this variable to your 2D and 3D scatter plots, as an optional parameter `c=colors`. Re-run your assignment and answer the final question below.

Lab Questions (Continued)

(2/2 points)
Reset your 'n_neighbors' if you changed it from 6. After adding in the additional images from the 32_i dataset, do examine your 2D and 3D scatter plots again. Have the new samples altered the shape of your original (blue) manifold?

○ No, not in the slightest

● Only *very slightly*... ✓

○ It looks like a completely different shape

EXPLANATION

The manifold shape should only have been altered ever so slightly, barely noticeably. The new samples sort of act as noise, but do not short circuit the original manifold.

What is the arrangement of the newly added, red samples?

● Isomap rendered the result as a straight line, intersecting the original manifold ✓

○ They are completely sporadic compared to the original manifold, so no real pattern exist

○ They are aligned in a smaller circle, intersecting the original manifold

EXPLANATION

The new samples intersect the original geometry at one of the inflection points. If you're one easily tugged by curiosity, modify the 'Plot2D' code from the previous assignments to figure out which of the original, blue samples is 'closest' to the new 'red' samples!

You have used 1 of 2 submissions

© All Rights Reserved

© edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY OPENedX

Download on the App Store

ANDROID APP ON Google play