Project: scikit-learn

2016-04-15T11:45:35Z joelkuiper 106comments

t-SNE fails with array must not contain infs or NaNs (OSX specific)

Web url: https://github.com/scikit-learn/scikit-learn/issues/6665

API url: https://api.github.com/repos/scikit-learn/scikit-learn/issues/6665

```

Darwin-15.0.0-x86\_64-i386-64bit

('Python', '2.7.11 |Anaconda custom (x86\_64)| (default, Dec 6 2015, 18:57:58) \n[GCC 4.2.1 (Apple Inc. build 5577)]')

('NumPy', '1.11.0')

('SciPy', '0.17.0')

('Scikit-Learn', '0.17.1')

```

When trying to run a t-SNE

```

proj = TSNE().fit\_transform(X)

ValueError: array must not contain infs or NaNs

```

However

```

np.isfinite(X).all() # True

np.isnan(X).all() # False

np.isinf(X).all() # False

```

Full Stack Trace:

```

ValueError Traceback (most recent call last)

<ipython-input-16-c25f35fd042c> in <module>()

----> 1 plot(X, y)

<ipython-input-1-72bdb7124d13> in plot(X, y)

74

75 def plot(X, y):

---> 76 proj = TSNE().fit\_transform(X)

77 scatter(proj, y)

/Users/joelkuiper/anaconda/lib/python2.7/site-packages/sklearn/manifold/t\_sne.pyc in fit\_transform(self, X, y)

864 Embedding of the training data in low-dimensional space.

865 """

--> 866 embedding = self.\_fit(X)

867 self.embedding\_ = embedding

868 return self.embedding\_

/Users/joelkuiper/anaconda/lib/python2.7/site-packages/sklearn/manifold/t\_sne.pyc in \_fit(self, X, skip\_num\_points)

775 X\_embedded=X\_embedded,

776 neighbors=neighbors\_nn,

--> 777 skip\_num\_points=skip\_num\_points)

778

779 def \_tsne(self, P, degrees\_of\_freedom, n\_samples, random\_state,

/Users/joelkuiper/anaconda/lib/python2.7/site-packages/sklearn/manifold/t\_sne.pyc in \_tsne(self, P, degrees\_of\_freedom, n\_samples, random\_state, X\_embedded, neighbors, skip\_num\_points)

830 opt\_args['momentum'] = 0.8

831 opt\_args['it'] = it + 1

--> 832 params, error, it = \_gradient\_descent(obj\_func, params, \*\*opt\_args)

833 if self.verbose:

834 print("[t-SNE] Error after %d iterations with early "

/Users/joelkuiper/anaconda/lib/python2.7/site-packages/sklearn/manifold/t\_sne.pyc in \_gradient\_descent(objective, p0, it, n\_iter, objective\_error, n\_iter\_check, n\_iter\_without\_progress, momentum, learning\_rate, min\_gain, min\_grad\_norm, min\_error\_diff, verbose, args, kwargs)

385 for i in range(it, n\_iter):

386 new\_error, grad = objective(p, \*args, \*\*kwargs)

--> 387 grad\_norm = linalg.norm(grad)

388

389 inc = update \* grad >= 0.0

/Users/joelkuiper/anaconda/lib/python2.7/site-packages/scipy/linalg/misc.pyc in norm(a, ord, axis, keepdims)

127 """

128 # Differs from numpy only in non-finite handling and the use of blas.

--> 129 a = np.asarray\_chkfinite(a)

130

131 # Only use optimized norms if axis and keepdims are not specified.

/Users/joelkuiper/anaconda/lib/python2.7/site-packages/numpy/lib/function\_base.pyc in asarray\_chkfinite(a, dtype, order)

1020 if a.dtype.char in typecodes['AllFloat'] and not np.isfinite(a).all():

1021 raise ValueError(

-> 1022 "array must not contain infs or NaNs")

1023 return a

1024

ValueError: array must not contain infs or NaNs

```

-------------------------------------------------------------------------

2016-04-15T12:18:25Z joelkuiper

Same with ('Scikit-Learn', '0.18.dev0')

-------------------------------------------------------------------------

2016-04-17T07:11:03Z KeyKy

Do you mind sharing your data X with me?

-------------------------------------------------------------------------

2016-04-17T08:56:36Z joelkuiper

Sure, where and in what format would you like it?

> On 17 Apr 2016, at 09:11, Â∫∑Ê¥ã notifications@github.com wrote:

>

> Do you mind sharing your data X with me?

>

> ‚Äî

> You are receiving this because you authored the thread.

> Reply to this email directly or view it on GitHub https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-210968577

-------------------------------------------------------------------------

2016-04-17T09:06:51Z KeyKy

My email is 370846270@qq.com

As i know, there is a function numpy.save for saving an array to a binary file in .npy format~~

-------------------------------------------------------------------------

2016-04-18T01:10:06Z KeyKy

I test your data in ubuntu 14.04 LTS with

Python==2.7.6

scikit-learn==0.17.1

numpy==1.8.2

scipy==0.13.3

It is fine and doesn't raise the ValueError. The test code is:

`import numpy

a = numpy.load('/root/test.npy')

print a.shape

print numpy.isnan(a).all() #False

print numpy.isfinite(a).all() #True

print numpy.isinf(a).all() #False

from sklearn.manifold import TSNE

proj = TSNE().fit\_transform(a) #[[ 2.35503527e+00 1.15976751e+01] .... [ 3.29832591e+00 8.98212513e+00]]

print proj`

---

Then i upgrade numpy, scipy to 1.11.0, 0.17.0 and test with the same code and it also doesn't raise any error.

-------------------------------------------------------------------------

2016-05-11T05:29:51Z ivan-krukov

Reproduced for 3.5 with anaconda under OS X El Capitan.

```

Darwin 15.4.0

Python 3.5.1 :: Anaconda custom (x86\_64)

numpy 1.10.4

scipy 0.17.0

scikit-learn 0.17.1

```

Example run:

``` python

import random

from sklearn.manifold import TSNE

random.seed(1)

a = np.random.uniform(size=(100,20))

TSNE(n\_components=2).fit\_transform(a)

```

-------------------------------------------------------------------------

2016-05-11T06:10:09Z jnothman

Thanks @ivan-krukov, but I'm failing to replicate in Python 3.3. Will try 3.5

-------------------------------------------------------------------------

2016-05-11T16:08:49Z ivan-krukov

This does not apply to `linux` (4.4.0-21, Ubuntu 16.04) with the same packages under 3.5.

-------------------------------------------------------------------------

2016-05-11T22:28:28Z jnothman

I'm on El-Capitan, but I'm failing to get a Python 3.5 installation up and running.

-------------------------------------------------------------------------

2016-06-01T15:17:05Z dcbb

Is there any update on this?

I have the issue on a dataset of mine, on Anaconda, Py 3.5, sklearn 0.17.1, OSX El Capitan.

I can reproduce the error with the example provided by @ivan-krukov .

-------------------------------------------------------------------------

2016-06-02T08:46:34Z youyanggu

Same issue. Python 2.7.6 on OS X El Capitan on 0.17. Tried the same code on Linux using Python 2.7.6 and 0.17, and it works.

-------------------------------------------------------------------------

2016-06-08T11:13:35Z edevil

Same issue.

OSX El Capitan Python 3.5.1

scikit-learn==0.17.1

scipy==0.17.1

-------------------------------------------------------------------------

2016-06-13T17:20:01Z Ekliptor

I have the same problem and would really appreciate a fix (or workaround?)

System Version: OS X 10.11.5

Python 3.5.1 :: Anaconda 4.0.0 (x86\_64)

numpy.version.version 1.11.0

scipy.version 0.17.1

sklearn.\*\*version\*\* 0.17.1

I can also reproduce the bug with the code sample from ivan-krukov

-------------------------------------------------------------------------

2016-06-16T13:42:38Z lucienevans

Same issue on OS X EI Capitan using Python 3.5

-------------------------------------------------------------------------

2016-06-29T19:59:31Z Concomitant

System Version: OS X 10.11.5

Python 3.5.1 :: Continuum Analytics, Inc.

numpy.\*\*version\*\* 1.11.1

scipy.\*\*version\*\* 0.16.0

sklearn.\*\*version\*\* 0.17.1

Same problem. Though I have noticed that \*\*it only occurs for a subset of my dataset and not with the whole thing\*\*. That is, if I do TSNE on the whole data set it \*\*works\*\*, if I do it on a reduced set it does not.

-------------------------------------------------------------------------

2016-06-29T20:06:13Z Concomitant

O\_o;; This just in, if I repeat the same 'broken' subset that doesn't work(by means of list\*10) then it works. Multiplying each individual vector by 10 doesn't work, but duplicating the date does. just doubling the length of the list is insufficient. Maybe this is some kind of degrees of freedom check run amok?

-------------------------------------------------------------------------

2016-06-30T11:44:54Z lesteve

@ivan-krukov I bit the bullet today and installed an El Capitan VM. Unfortunately I can not reproduce your problem.

@Concomitant can you reproduce the error on the stand-alone example given in https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-218365487?

-------------------------------------------------------------------------

2016-06-30T12:24:47Z lesteve

> I'm on El-Capitan, but I'm failing to get a Python 3.5 installation up and running.

@jnothman it doesn't seem to be happening only on Python 3.5 so if you could try to reproduce with Python 2.7 (snippet: https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-218365487) that would be great.

-------------------------------------------------------------------------

2016-06-30T15:55:05Z Concomitant

@lesteve I can reproduce the issue.

```

import numpy as np

import random

from sklearn.manifold import TSNE

random.seed(1)

a = np.random.uniform(size=(100,20))

TSNE(n\_components=2).fit\_transform(a)

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

File "/Users/dshank/miniconda3/envs/python3/lib/python3.5/site-packages/sklearn/manifold/t\_sne.py", line 866, in fit\_transform

embedding = self.\_fit(X)

File "/Users/dshank/miniconda3/envs/python3/lib/python3.5/site-packages/sklearn/manifold/t\_sne.py", line 777, in \_fit

skip\_num\_points=skip\_num\_points)

File "/Users/dshank/miniconda3/envs/python3/lib/python3.5/site-packages/sklearn/manifold/t\_sne.py", line 832, in \_tsne

params, error, it = \_gradient\_descent(obj\_func, params, \*\*opt\_args)

File "/Users/dshank/miniconda3/envs/python3/lib/python3.5/site-packages/sklearn/manifold/t\_sne.py", line 387, in \_gradient\_descent

grad\_norm = linalg.norm(grad)

File "/Users/dshank/miniconda3/envs/python3/lib/python3.5/site-packages/scipy/linalg/misc.py", line 115, in norm

a = np.asarray\_chkfinite(a)

File "/Users/dshank/miniconda3/envs/python3/lib/python3.5/site-packages/numpy/lib/function\_base.py", line 1033, in asarray\_chkfinite

"array must not contain infs or NaNs")

ValueError: array must not contain infs or NaNs

```

Following the same code, however:

```

>>> a = np.random.uniform(size=(10000,20))

>>> TSNE(n\_components=2).fit\_transform(a)

array([[ 3.25766047e+11, -2.74708004e+11],

[ 2.43498802e+11, -7.68189047e+10],

[ -6.00107639e+09, -1.13548763e+11],

...,

[ 3.02794039e+10, 6.64402020e+11],

[ 2.55855781e+10, 5.67932400e+10],

[ 1.42040378e+11, -7.55188994e+10]])

```

Bizarre.

-------------------------------------------------------------------------

2016-07-05T14:09:56Z ogrisel

I cannot reproduce either with python 3.5.1, numpy 1.11.1, scipy 0.17.1 and scikit-learn 0.17.1 from miniconda (with MKL) on a virtualbox with OSX El Capitan. I will try on a real mac hardware later.

-------------------------------------------------------------------------

2016-07-05T14:20:17Z ogrisel

Also @joelkuiper and @Concomitant can you please check that you can reproduce the problem on the current state of the scikit-learn master branch?

-------------------------------------------------------------------------

2016-07-05T14:49:42Z nelson-liu

@lesteve and others I cannot reproduce the error with the [snippet posted earlier](https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-218365487) on the latest master with python 2.7.

System info:

```

Darwin-15.0.0-x86\_64-i386-64bit

('Python', '2.7.10 (v2.7.10:15c95b7d81dc, May 23 2015, 09:33:12) \n[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)]')

('NumPy', '1.11.0')

('SciPy', '0.17.0')

('Scikit-Learn', '0.18.dev0')

```

-------------------------------------------------------------------------

2016-07-06T07:26:04Z ogrisel

I tried again on a real mac running OSX El Capitan 10.11.3 (with anaconda's latest numpy scipy and scikit-learn, same setting as reported by @Concomitant in https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-229703129) but could not reproduce the problem either (tried running the snippet several times).

What is weird though it that the despite the `np.random.seed(1)` line I get different results for the output of `fit\_transform`. This might be a bug in itself.

-------------------------------------------------------------------------

2016-07-06T12:19:09Z ogrisel

> What is weird though it that the despite the np.random.seed(1) line I get different results for the output of fit\_transform. This might be a bug in itself.

Actually I read @Concomitant's code snippet too quickly: instead of `random.seed(1)` it should be `np.random.seed(1)` otherwise the numpy RNG is not reseeded appropriately and one cannot get deterministic results.

-------------------------------------------------------------------------

2016-07-06T12:20:08Z ogrisel

Also I now realized that I read the whole discussion too quickly and that the bug only happens with python 2.7. Will try again.

-------------------------------------------------------------------------

2016-07-06T12:31:54Z ogrisel

I cannot reproduce either with python 2.7.12 from conda on OSX 10.11.3 either.

Actually @Ekliptor can reproduce the issue with python 3.5.1 from conda so it's probably not related to the version of Python either. Maybe it depends on the minor version of OSX. Will upgrade and retry.

-------------------------------------------------------------------------

2016-07-06T13:00:53Z ogrisel

I cannot replicate either with OSX 10.11.5. I tried both with Python 2.7.12 and 3.5.2 installed with conda along with numpy 1.11.1, scipy 0.17.1 and scikit-learn 0.17.1.

I don't know what to do. If one of you can reproduce the problem, please try to find a numpy random seed that trigger the issue (using `np.random.seed(my\_seed)` instead of `random.seed(1)` in the above snippet) and communicate the value here (along with the version of OSX and you python packages).

-------------------------------------------------------------------------

2016-07-11T11:16:39Z Ekliptor

I can confirm the issue is fixed with the latest version. I can not reproduce it anymore as before.

I only updated numpy:

numpy.version.version 1.11.1

To all people working with Tensorflow I can add:

When I try to plot a very small sample (< 200 points) I sometimes still run into this error. After increasing the sample size I pass into tsne.fit\_transform() it always works.

-------------------------------------------------------------------------

2016-07-11T11:59:00Z ogrisel

Thanks @Ekliptor for checking that it works with scikit-learn master. @joelkuiper and @Concomitant do you confirm that scikit-learn master also work for you? If so we can close this issue.

-------------------------------------------------------------------------

2016-07-11T13:50:44Z Concomitant

I installed master, the code snippet runs cleanly now.

-------------------------------------------------------------------------

2016-07-28T20:11:12Z amueller

seems to work for everybody now. closing.

-------------------------------------------------------------------------

2016-08-01T19:09:27Z dmyersturnbull

Sorry, but I still get this on Python 3.5.1, scikit 0.17, scikit-learn 0.18 (commit 9e913c04d748), and Numpy 1.11.1 on Mac OS 10.11.5.

-------------------------------------------------------------------------

2016-08-02T03:00:46Z lesteve

@dmyersturnbull do you get the error when running the snippet from https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-218365487?

-------------------------------------------------------------------------

2016-08-02T18:08:16Z dmyersturnbull

@lesteve I did with that exact snippet, yes. However, I no longer get it after clearing my Anaconda installation and reinstalling from scratch with Python 3.5.2.

-------------------------------------------------------------------------

2016-08-09T18:12:18Z jna29

I get the same problem with Python 3.5.2, scikit-learn 0.17.1, scipy 0.17.1, numpy 1.11.1 on Mac OS X El Capitan 10.11.3. It works when I have more than 2100 points but fails for lower values.

-------------------------------------------------------------------------

2016-08-24T08:09:09Z lucidyan

> I get the same problem with Python 3.5.2, scikit-learn 0.17.1, scipy 0.17.1, numpy 1.11.1 on Mac OS X El Capitan 10.11.3. It works when I have more than 2100 points but fails for lower values.

Analogically \*\*fails for low points' values\*\*

Reopen, please

-------------------------------------------------------------------------

2016-08-31T12:43:21Z pbnsilva

I am getting the same problem on OS X 10.11.6, python 3.5.1, sklearn 0.17.1 and numpy 1.11.1 .

On this dataset: https://dl.dropboxusercontent.com/u/103591/vals.out (with np.savetxt)

-------------------------------------------------------------------------

2016-08-31T14:29:27Z lesteve

> Analogically fails for low points' values

@Lucidyan I don't understand what you mean by that.

> I am getting the same problem on OS X 10.11.6, python 3.5.1, sklearn 0.17.1 and numpy 1.11.1 .

> On this dataset: https://dl.dropboxusercontent.com/u/103591/vals.out (with np.savetxt)

@pbnsilva can you try this snippet posted below ? You may need to run it multiple times because unfortunately the seed is not set appropriately (you need to use `np.random.seed` rather than `random.seed`).

``` py

import random

from sklearn.manifold import TSNE

random.seed(1)

a = np.random.uniform(size=(100,20))

TSNE(n\_components=2).fit\_transform(a)

```

Bonus points if you can find a seed argument to `np.random.seed` and a `random\_state` argument to `TSNE` that makes the snippet deterministic.

Alternatively some people reported that this bug was fixed in master. Could you try to build scikit-learn master to see whether the problem disappears ?

-------------------------------------------------------------------------

2016-09-02T13:04:18Z lucidyan

@lesteve I meant that I get the same error with a small number of instances, with the same system parameters (Python 3.5.2, scikit-learn 0.17.1, scipy 0.17.1, numpy 1.11.1 on Mac OS X El Capitan 10.11.3)

> @pbnsilva can you try this snippet posted below ? You may need to run it multiple times because unfortunately the seed is not set appropriately (you need to use np.random.seed rather than random.seed).

I tried it, and it fails with X\_SIZE <= 1750 (Y\_SIZE=20, n\_components=2 became constants). if I start to change the constants (increase) with fixed X\_SIZE=1750, it fails too.

-------------------------------------------------------------------------

2016-09-06T07:07:32Z lesteve

@Lucidyan could you try the same snippet with scikit-learn master and see whether it fails too ?

-------------------------------------------------------------------------

2016-09-20T07:49:44Z act65

yea not working for me (numpy 1.11.1, El capitan.10.11, sklearn 0.17.1, python 3.5.2) annoyingly it has broken old code that did work. what did you guys change...?

-------------------------------------------------------------------------

2016-09-20T08:26:09Z lesteve

@act65 we are more than keen to get to the bottom of this but we haven't been able to reproduce and it seems like we are getting mixed reports from users so far unfortunately.

So if you haven't already (unfortunately we are not mind readers and "not working for me" does not tell us what you tried) could you try to run the snippet mentioned above in https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-243782185. Try to run it multiple times just in case because the random seed is not set properly and there may be some randomness left in the snippet.

Then what would be really great if you could try with the 0.18 release candidate which is straightforward to install (highly recommended to do it in a separate virtualenv or conda env):

```

pip install --pre scikit-learn -U

```

\*\*Edited: 0.18 has been released so you can just use (no need to use `--pre`):\*\*

```

pip install scikit-learn -U

```

and re-run the snippet to see whether it is fixed in 0.18 as some users have reported in this thread already.

0.18 is going to be released in a few weeks if not days so you know what you have to do if you want to help us to get to the bottom of this before the release ;-).

-------------------------------------------------------------------------

2016-09-20T08:53:22Z act65

yea my bad, should have been clearer. (I had tried roughly the same thing others had, just on MNIST).

anyway, it works! thanks :)

`pip install --pre scikit-learn -U` fixed it

-------------------------------------------------------------------------

2016-09-20T09:07:45Z lesteve

OK, thanks for reporting back and great to hear that this is fixed for you in the 0.18 release candidate ! This seems to match what other have reported when they say it was fixed in master.

Just for completeness though, it is recommended to stick to released versions for production code, so you may need to wait a little bit more until the 0.18 release is out.

-------------------------------------------------------------------------

2016-09-20T19:57:41Z lucidyan

@lesteve

I tried the [snippet](https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-243782185) on version 0.18rc2, installed by

> pip install --pre scikit-learn -U

And it seems working! Cheers!

-------------------------------------------------------------------------

2016-09-21T06:22:18Z lesteve

Thanks @Lucidyan for giving it a try.

-------------------------------------------------------------------------

2016-10-31T20:22:22Z zhongyuk

Sorry, I'm still getting this error with above code snippet after upgrading to scikit-learn 0.18 (`pip install --pre scikit-learn -U`) in conda env.

Here is my system info:

OS X El Capitan Version 10.11.4

Python 2.7.12

sklearn 0.18 (got the the same error on sklearn 0.17.1 as well)

numpy 1.11.1 ( got the same error on numpy 1.11.2 as well)

scipy 0.18.1

However, I ran the same code snippet on Linux system, I didn't get an error.

The system info of the Linux system is:

Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-91-generic x86\_64)

Python 2.7.6

sklearn 0.18

numpy 1.11.2

scipy 0.13.3

-------------------------------------------------------------------------

2016-11-02T07:51:37Z lesteve

Try uninstalling and reinstalling numpy, scipy and scikit-learn. If that still fails, try in a different virtualenv (or conda environment if you are using conda) to make sure something is not wrong in your Python environment.

-------------------------------------------------------------------------

2016-11-03T15:04:59Z rasbt

Still get the same error (`ValueError: array must not contain infs or NaNs`) in sklearn 0.18 (`0.18-np111py35\_0`) via conda. The pip wheels seem to work fine though!

-------------------------------------------------------------------------

2016-11-03T16:02:13Z lesteve

> Still get the same error (ValueError: array must not contain infs or NaNs) in sklearn 0.18 (0.18-np111py35\_0) via conda. The pip wheels seem to work fine though!

Hmmm interesting ... could you try using conda packages without mkl, i.e. something like `conda create -n sklearn\_nomkl python scikit-learn nomkl` so we can see whether that is a MKL vs openblas thing?

Also bonus points if you can provide a snippet reproducing the problem with a fixed random seed (i.e. using `np.random.RandomState(some\_int)`) that can be used as a reference snippet going forward. Up until now the snippet we have is non-deterministic (`random.seed` is used and has no influence of numpy.random seed).

-------------------------------------------------------------------------

2016-11-03T16:26:24Z rasbt

Sure, no problem. This may be be a BLAS problem indeed, the `conda create -n sklearn\_nomkl python scikit-learn nomkl` env works fine.

Regarding the snippet ... this gets interesting. E.g.,

``` python

from sklearn.manifold import TSNE

from sklearn.model\_selection import train\_test\_split

from sklearn.datasets import load\_digits

digits = load\_digits()

X\_train, X\_test, y\_train, y\_test = train\_test\_split(digits.data,

digits.target,

random\_state=1)

tsne = TSNE(random\_state=1)

digits\_tsne\_train = tsne.fit\_transform(X\_train)

```

reproduces the problem on my machine. However, when I replace `digits\_tsne\_train = tsne.fit\_transform(X\_train)` by `digits\_tsne\_train = tsne.fit\_transform(digits.data)` it seems to be fine. Would be good to find a more light-weight example maybe, to add this particular case to the travis tests.

EDIT: Same is true for iris. `iris.data` works in fit\_transform, a splitted dataset (`X\_train`) does not. Maybe there's sth funny going on in `train\_test\_split`. However, both `X\_train` and `iris.data` seem to be float 64 arrays ...

-------------------------------------------------------------------------

2016-11-03T16:49:30Z lesteve

What about the snippet from https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-243782185, you didn't find a way to make it deterministic and still fail on your machine?

-------------------------------------------------------------------------

2016-11-03T17:49:28Z rasbt

The snippet

``` python

import numpy as np

from sklearn.manifold import TSNE

np.random.seed(1)

a = np.random.uniform(size=(100, 20))

TSNE(n\_components=2, random\_state=1).fit\_transform(a)

```

reproduces the error (but it works fine on the nomkl env)

-------------------------------------------------------------------------

2016-11-03T21:30:51Z lesteve

OK thanks a lot for this, at least we have a deterministic snippet now. For the record, can you post the output of this snippet:

``` python

import platform; print(platform.platform())

import sys; print("Python", sys.version)

import numpy; print("NumPy", numpy.\_\_version\_\_)

import scipy; print("SciPy", scipy.\_\_version\_\_)

import sklearn; print("Scikit-Learn", sklearn.\_\_version\_\_)

```

Also, just for the sake of sanity, can you make sure you can reproduce the problem in a fresh conda environment.

To be honest, I am not sure where we go from this. I haven't tried since but I was not able to reproduce on an El Capital Virtual box, @ogrisel could not reproduce either on a OSX laptop, so at the time he said there might be some hardware-specific problem involved.

-------------------------------------------------------------------------

2016-11-03T21:55:55Z rasbt

Sure,

the machine that causes this problem:

```

Darwin-16.1.0-x86\_64-i386-64bit

Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul 2 2016, 17:52:12)

[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)]

NumPy 1.11.2

SciPy 0.18.1

Scikit-Learn 0.18

```

(tested it in a fresh conda environment)

> there might be some hardware-specific problem involved.

I think you may be onto sth! I tried it on my other mac, and it works fine there. The only difference is to the output above it is running on an older kernel (`Darwin-15.6.0-x86\_64-i386-64bit`). Haven't updated the second mac to macOS Sierra yet, which is running on the former machine that has this problem. Could be OS-related. I will upgrade the second machine to Sierra in the next month or so (I am in the middle of a project and don't want to break things), but I can let you know if the update to Sierra leads to this issue on the second machine (or maybe someone else with macOS Sierra could test it so that we now if it is an OS thing)

-------------------------------------------------------------------------

2016-11-03T23:03:02Z lesteve

Given that the problem has been reported on different OSX versions, I kind of doubt this is only a OSX version issue. IIRC @ogrisel's hunch was that it was CPU architecture related.

Another (more time-intensive) way to debug this problem would be to track down where the NaNs appear in the code.

-------------------------------------------------------------------------

2016-11-03T23:17:19Z rasbt

> IIRC @ogrisel's hunch was that it was CPU architecture related.

Hm, how would the conda scikit-learn version differ from the pip wheels? Because the latter seem to work on the same machine. Maybe it's somehow related to conda

-------------------------------------------------------------------------

2016-11-04T00:14:05Z rasbt

> Another (more time-intensive) way to debug this problem would be to track down where the NaNs appear in the code.

I noticed that the gradient in https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/manifold/t\_sne.py#L387 explodes, until it becomes `-inf` in one position after the 25th iteration in the https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/manifold/t\_sne.py#L386 for-loop

``` python

...

0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00

0.00000000e+00 0.00000000e+00 6.06587795e+32 -1.10699515e+33

-1.55245133e+34 inf -1.52569936e+33 -3.43926080e+33

-1.92332051e+32 -2.73996151e+32 -2.57570880e+33 -3.64962271e+33

...

```

On the other machine (the one that works fine), the gradients are all < 0 after the same iteration. So, somehow the `\_gradient\_descent` function doesn't work properly (maybe due to some BLAS thing).

-------------------------------------------------------------------------

2016-11-04T06:18:18Z lesteve

> Hm, how would the conda scikit-learn version differ from the pip wheels? Because the latter seem to work on the same machine. Maybe it's somehow related to conda

The pip wheels are using OpenBLAS and you don't have the problem when using OpenBLAS with conda (through the `nomkl` trick) so this does look like a MKL problem, which on top of that is likely CPU-specific.

-------------------------------------------------------------------------

2016-11-04T07:20:38Z lesteve

Great job debugging the issue by the way! Bonus points if you manage to further isolate the problem (e.g. by pickling the data before the iteration `inf` appear). The problem arises very likely in some cython code in sklearn/manifold/\_barnes\_hut\_tsne.pyx.

-------------------------------------------------------------------------

2016-11-21T22:43:19Z luisatlive

Shouldn't this issue be re-opened given the latest findings? I hit it as well and also managed to get past it with the nomkl trick, but feels like an active bug vs. a closed one, no?

Others that have been hitting this: https://discussions.udacity.com/t/assignment-5-error-in-the-main-code-valueerror-array-must-not-contain-infs-or-nans/178187/7

-------------------------------------------------------------------------

2016-11-22T07:32:58Z lesteve

You are right, reopening. This one is a serious one, seems hardware specific and none of the core devs could reproduce it. The only way this can get fixed is if people having the issue invest some time in debugging the problem further.

-------------------------------------------------------------------------

2016-11-22T17:55:54Z rasbt

> Great job debugging the issue by the way! Bonus points if you manage to further isolate the problem (e.g. by pickling the data before the iteration inf appear). The problem arises very likely in some cython code in sklearn/manifold/\_barnes\_hut\_tsne.pyx.

I am happy to look into it further in December after all the November deadlines ... However, even this can be further isolated, I am curious if there's a fix for such a hardware-specific problem. Maybe, until this is fully resolved, it may be worthwhile to raise a more specific exception/warning if the gradient contains infs with a note about this problem?

-------------------------------------------------------------------------

2016-11-22T18:07:38Z zhongyuk

I just created a new conda virtualenv and built a devp version of sklearn from the source code freshly forked from the sciki-learn master branch, the error disappeared. Is the devp sklearn built from source code using OpenBLAS instead of MKL?

-------------------------------------------------------------------------

2016-11-22T23:04:33Z lesteve

> I am happy to look into it further in December after all the November deadlines ...

Sounds great, thanks a lot !

> However, even this can be further isolated, I am curious if there's a fix for such a hardware-specific problem.

Not sure about a fix, one hope would be if we can change our cython code to work-around problem once we have isolated it. Also it could well be an openblas issue and that would be great reporting it upstream, especially since wheels use openblas.

> Maybe, until this is fully resolved, it may be worthwhile to raise a more specific exception/warning if the gradient contains infs with a note about this problem?

Adding some advice to the error message (only on OS X), sounds like a good idea, but I am not sure what it should say, maybe "consider using conda and install scikit-learn with MKL" or something like this.

-------------------------------------------------------------------------

2016-11-22T23:13:42Z lesteve

> Is the devp sklearn built from source code using OpenBLAS instead of MKL?

@zhongyuk depends which library you have installed. One way to know once you have built scikit-learn from source is to run the equivalent of `ldd` (Google seems to say `otool -L`) on `sklearn/cluster/\_k\_means.so` (name will be different if you are using Python 3, i.e. something like `sklearn/cluster/\_k\_means.cpython-35m-x86\_64-linux-gnu.so `). On my Ubuntu machine for example, I get this:

```

sklearn/cluster/\_k\_means.so:

linux-vdso.so.1 => (0x00007ffc2312a000)

libmkl\_intel\_lp64.so => /home/lesteve/miniconda3/envs/py27/lib/libmkl\_intel\_lp64.so (0x00007fadc2865000)

libmkl\_intel\_thread.so => /home/lesteve/miniconda3/envs/py27/lib/libmkl\_intel\_thread.so (0x00007fadc0ee4000)

libmkl\_core.so => /home/lesteve/miniconda3/envs/py27/lib/libmkl\_core.so (0x00007fadbf483000)

libiomp5.so => /home/lesteve/miniconda3/envs/py27/lib/libiomp5.so (0x00007fadbf139000)

libpthread.so.0 => /lib/x86\_64-linux-gnu/libpthread.so.0 (0x00007fadbeeeb000)

libm.so.6 => /lib/x86\_64-linux-gnu/libm.so.6 (0x00007fadbebe1000)

libpython2.7.so.1.0 => /home/lesteve/miniconda3/envs/py27/lib/libpython2.7.so.1.0 (0x00007fadbe7fa000)

libc.so.6 => /lib/x86\_64-linux-gnu/libc.so.6 (0x00007fadbe431000)

libdl.so.2 => /lib/x86\_64-linux-gnu/libdl.so.2 (0x00007fadbe22c000)

libgcc\_s.so.1 => /lib/x86\_64-linux-gnu/libgcc\_s.so.1 (0x00007fadbe016000)

/lib64/ld-linux-x86-64.so.2 (0x0000563bdeda1000)

libutil.so.1 => /lib/x86\_64-linux-gnu/libutil.so.1 (0x00007fadbde12000)

```

So you can see from the third line, that it is using MKL.

-------------------------------------------------------------------------

2016-11-22T23:18:44Z rasbt

> but I am not sure what it should say, maybe "consider using conda and install scikit-learn with MKL" or something like this.

I just wanted to write that I think you got it flipped: the wheels worked find and the issue only occured when I was using it via conda with MKL ... Now, I think I have good news in some way: I just wanted to rerun the above example that previously caused this issue to confirm

```python

import numpy as np

from sklearn.manifold import TSNE

np.random.seed(1)

a = np.random.uniform(size=(100, 20))

TSNE(n\_components=2, random\_state=1).fit\_transform(a)

```

and I am no longer getting this problem. I remember that I reininstalled miniconda the other week due to some other problems. Do you think it could be related to some issue in the old conda? Would be great if some other people who had this issue could maybe also try updating/reinstalling conda and check if that solves the problem for them. Meanwhile, I will try to see if I can find an old backupstate to find out which conda version I had installed previously. (right now, I have conda 4.2.12)

-------------------------------------------------------------------------

2016-11-23T00:41:04Z zhongyuk

Just wanna say that I ran `otool -L` on `sklearn/manifold/\_barnes\_hut\_tsne.so` (I assume this is the t\_sne.py compiled file?), it seems like it's indeed using BLAS. And the one which threw error seems to use MKL..

The conda version I have is 4.2.13, both the env which throws the error and the env with source built sklearn (which does not throw error) are inside conda.

-------------------------------------------------------------------------

2016-11-23T01:17:51Z rasbt

Hm, interesting, so it's not a conda issue after all then ... Curious why it works for me now :/

(all I can think that has changed (except for reinstalling conda) was rebooting :P)

-------------------------------------------------------------------------

2016-11-23T09:07:09Z lesteve

> I just wanted to write that I think you got it flipped: the wheels worked find and the issue only occured when I was using it via conda with MKL

Yeah, sorry about that. I'll edit the issue title to try to remember it right for next time.

> Hm, interesting, so it's not a conda issue after all then ... Curious why it works for me now :/

(all I can think that has changed (except for reinstalling conda) was rebooting :P)

Hmmm, random guess maybe the mkl version, although if I believe the output of `conda info mkl` the latest mkl version (11.3.3) is from 2016-05-13.

-------------------------------------------------------------------------

2016-11-23T09:17:33Z lesteve

@zhongyuk try to build scikit-learn inside a conda env that uses mkl, I believe this should be enough for mkl to be picked up (probably a good idea in this case to do `make clean` and then `make in` to rebuild from scratch).

-------------------------------------------------------------------------

2016-11-24T04:37:26Z zhongyuk

@lesteve I built scikit-learn in two conda virtual environments from source code (branch 0.18 release), the one uses MKL indeed throws the error; the one uses libBLAS does not throw error.

The output running `otool -L` on `sklearn/manifold/\_barnes\_hut\_tsne.so` is here (in case MKL version gives you any clue?)

```@rpath/libmkl\_intel\_lp64.dylib (compatibility version 0.0.0, current version 0.0.0)

@rpath/libmkl\_intel\_thread.dylib (compatibility version 0.0.0, current version 0.0.0)

@rpath/libmkl\_core.dylib (compatibility version 0.0.0, current version 0.0.0)

@rpath/libiomp5.dylib (compatibility version 5.0.0, current version 5.0.0)

/usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1226.10.1)

-------------------------------------------------------------------------

2016-11-24T07:02:04Z lesteve

@zhongyuk great ! For completeness, can you post the output of `conda list '(mkl|cython|numpy|scipy)$'` (in your MKL conda environment)? While we are at it your CPU information (`sysctl -n machdep.cpu.brand\_string` according to Google) and your platform information (`python -c 'import platform; print(platform.platform())'`) would be great.

What would be really great is to continue where @rabst stopped and further isolate the problem:

https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-258311980

Since this is related to BLAS, my hunch is that something goes wrong in this [line](https://github.com/scikit-learn/scikit-learn/blob/46fc1be145bf952a916ef5abd06efec105105c2e/sklearn/manifold/\_barnes\_hut\_tsne.pyx#L660) causing the gradient to have some non-finite values.

-------------------------------------------------------------------------

2016-11-24T15:28:26Z zhongyuk

@lesteve Output of conda MKL environment info:

```

Cython 0.25.1 <pip>

mkl 11.3.3 0

numpy 1.11.1 <pip>

numpy 1.11.1 py27\_0

scipy 0.18.1 np111py27\_0

```

CPU info: `Intel(R) Core(TM) i5-5257U CPU @ 2.70GHz`

Platform info: `Darwin-15.4.0-x86\_64-i386-64bit`

I'll look into the gradient exploding issue that @rabst found in the other comment and the line you pointed out sometime this week and/or next week, I'll keep everyone updated with any interesting findings.

-------------------------------------------------------------------------

2016-11-24T23:22:17Z rasbt

@zhongyuk If it helps, I have a very similar setup (can't reproduce the issue anymore since reinstalling miniconda), except that I have macOS Sierra instead of OS X El Capitan and that I have numpy 1.11.2 instead of 1.11.1.

-------------------------------------------------------------------------

2016-11-24T23:31:22Z zhongyuk

@rasbt hmm, I wonder if the problem goes away in Sierra... I don't want to upgrade OS yet b/z I thought I read somewhere that TensorFlow doesn't support Sierra yet (could be mistaken or no longer be true anymore since I don't remember where or how long ago I read it)? And I don't wanna break my projects with TF dependency

-------------------------------------------------------------------------

2016-11-24T23:35:06Z rasbt

@zhongyuk Hm, I think it's unlikely that it is related. Before I reinstalled miniconda, I also had the problem in macOS Sierra. PS: Tensorflow works fine for me on Sierra, but I only do CPU and prototyping on my macs so I don't know about GPU issues related to Sierra

-------------------------------------------------------------------------

2016-11-24T23:40:59Z zhongyuk

@rasbt hmm, that's good to know that TF works fine on Sierra. Do you wanna run `otool -L` on the `sklearn/manifold/\_barnes\_hut\_tsne.so` file in your platform to see which math library sklearn using underneath? At least that way we might know if the problem went away after reinstalling miniconda is fundamentally linked to math library?

-------------------------------------------------------------------------

2016-11-25T00:30:52Z rasbt

I am getting the following on `\_barnes\_hut\_tsne.cpython-35m-darwin.so`:

```

@rpath/libmkl\_intel\_lp64.dylib (compatibility version 0.0.0, current version 0.0.0)

@rpath/libmkl\_intel\_thread.dylib (compatibility version 0.0.0, current version 0.0.0)

@rpath/libmkl\_core.dylib (compatibility version 0.0.0, current version 0.0.0)

@rpath/libiomp5.dylib (compatibility version 5.0.0, current version 5.0.0)

/usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1197.1.1)

```

-------------------------------------------------------------------------

2016-11-25T00:39:54Z zhongyuk

@rasbt Hmm, that's really interesting. It's using MKL as well. I don't know enough about the math library to speculate what does this mean... @lesteve probably will be able to infer more from this?

I noticed that in my platform `libmkl\_intel\_lp64.dylib` is not loaded... Is it possible that caused the problem?

-------------------------------------------------------------------------

2016-11-25T00:51:10Z zhongyuk

WOW, yes! it is caused by `libmkl\_intel\_lp64.dylib` not loaded!!! I found [this thread on stack overflow](http://stackoverflow.com/questions/35392996/library-not-loaded-libmkl-intel-lp64-dylib-using-anaconda-on-os-x) and then ran `conda install --debug mkl`, then ran `otool -L` on the `sklearn/manifold/\_barnes\_hut\_tsne.so` file, then `libmkl\_intel\_lp64.dylib` loaded up, and ran the code snippet, error went away! Have five team work! @rasbt

If anyone else could check on their platform and see if the error went away after making sure `libmkl\_intel\_lp64.dylib` is loaded, that would be great!

@lesteve Since it does look like a lot of ppl has hit this problem, and it does look like it's related to (some version of ?) conda not extracting full MKL libraries (my understanding of the situation so far), even though it's not a scikit-learn bug, I do think either add some kind of remark or warning or error messages to (OS X) users would be nice? That way at least they can check if MKL lib is fully extracted in their platform and then fix it if it's not?

-------------------------------------------------------------------------

2016-11-25T03:01:08Z rasbt

@zhongyuk awesome, glad to hear that you were able to narrow it down! Hopefully, it's just the broken link/incomplete install of the `libmkl\_intel\_lp64.dylib` -- that would be awesome (in terms of knowing what's going on) :). That would also explain why it works for me now after re-installing Miniconda ... Would be great if someone else could try the "fix."

If the aforementioned `libmkl\_intel\_lp64.dylib` really caused this issue, the remaining question would be how to deal with that in scikit-learn. I mean, this "bug" is kind of hideous and it may be a bit tricky for folks to figure out that it's due to `libmkl\_intel\_lp64.dylib`. I probably wouldn't inject an additional "if gradient contains inf raise error + message" in the code in scikit-learn since it could be quite annoying performance-wise. However, i think that adding a note or comment in the installation and/or T-SNE docs would be a good idea.

-------------------------------------------------------------------------

2016-11-25T04:56:55Z zhongyuk

Just want to add a quick update: I had 2 virtual envs in conda both using MKL. One of them is equipped with `numpy 1.11.1` and the other is equipped with `numpy 1.11.2`. Running `otool -L` indicated that both of them somehow didn't have `libmkl\_intel\_lp64.dylib` loaded. After making sure `libmkl\_intel\_lp64.dylib` loaded up, the error disappeared in the virtual env with `numpy 1.11.2`. However, the error remained appearing in the env with `numpy 1.11.1`. After upgrading `numpy` to 1.11.2, I can no longer reproduce the error in either conda virtual environment. As it sounds complicated and the exact cause of the error is still obscure, I speculate it's probably a complication interweaved by incomplete MKL library loading and scikit-learn dependent libraries (possibly numpy?). (Although I haven't tried to create an virtualenv with MKL and `numpy 1.11.1` to see if this would reproduce the error.)

And I second @rasbt suggestion on adding some kind of note, comment or docs!

-------------------------------------------------------------------------

2016-11-25T19:19:32Z lesteve

@zhongyuk glad you got it fixed ! It seems that reinstalling packages with conda may help but I am afraid there doesn't seem to be a very clear picture of the cause of the problem :(.

-------------------------------------------------------------------------

2016-11-30T21:48:55Z amueller

This is a conda bug, right? Or did anyone experience the bug not using conda?

-------------------------------------------------------------------------

2016-11-30T23:28:07Z lesteve

I managed to find a way to reproduce I think by installing the numpy wheel and then scikit-learn via conda on top of it (got the hint from the `conda list` output in https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-262800762 where two `numpy` are listed).

```bash

conda create -n tmp python=3 -y

. activate tmp

pip install numpy -y

conda install scikit-learn -y

```

then execute the snippet from https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-262800762.

So it seems like this is happening when mixing numpy installed via pip and conda. In my book this is never a good idea to mix pip and conda for a given package but I guess this can happen without realizing it quite easily (for example you install a project that depends on numpy via pip, and then scikit-learn via conda).

Why this exactly happens I don't know ... and it seems to happen only on OSX by the way (i.e. not on my Ubuntu box).

-------------------------------------------------------------------------

2016-12-01T06:07:12Z lesteve

For anyone affected by this, this should fix it:

```bash

conda remove numpy --force -y

pip uninstall numpy -y

conda install numpy

```

Let me know if that doesn't work for you.

-------------------------------------------------------------------------

2016-12-01T21:43:19Z amueller

Thanks for the deep dive (again!) @lesteve

-------------------------------------------------------------------------

2016-12-02T10:17:43Z lesteve

I thought we would never get to the bottom of this one to be honest :) ! OK it's not quite the bottom but it's low enough as far as I am concerned.

I have to admit I would still like to understand what's happening within the numpy installed with both pip and conda ...

-------------------------------------------------------------------------

2017-02-24T01:35:41Z BerenLuthien

Hi

I tried two setups, where

- TSNE works well with one setup (where Tensorflow is de-activated, Python-3.x), however,

- TSNE does not work with the other setup (where Tensorflow is activated, Python 2.x).

\*\*The set up where TSNE works well:\*\*

Terminal:

```

Macbook:~ BG$ which jupyter

/Users/BG/anaconda/bin/jupyter

```

Jupyer notebook:

```

import sys

print (sys.version)

```

>>

```

3.5.2 |Anaconda 4.2.0 (x86\_64)| (default, Jul 2 2016, 17:52:12)

[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)]

```

Note: I tried

```

conda remove numpy --force -y

pip uninstall numpy -y

conda install numpy

```

to make TSNE work well with Tensorflow deactivated.

However, with the new setup below (where I have to use Tensorflow), this does not work any more.

‚Äî‚Äî-‚Äî‚Äî‚Äî-‚Äî‚Äî‚Äî-‚Äî‚Äî‚Äî-‚Äî‚Äî‚Äî-‚Äî‚Äî‚Äî-‚Äî‚Äî‚Äî-‚Äî‚Äî‚Äî-

\*\*The set up where TSNE does not work:\*\*

\*\*Terminal:\*\*

```

Macbook:~$ source activate tensorflow

(tensorflow) Macbook:~$ which jupyter

/Users//anaconda/envs/tensorflow/bin/jupyter

(tensorflow) Macbook:~$

```

\*\*Jupyer notebook:\*\*

```

import sys

print (sys.version)

```

>>

```

2.7.13 |Continuum Analytics, Inc.| (default, Dec 20 2016, 23:05:08)

[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)]

```

\*\*Error\*\*:

`ValueError: array must not contain infs or NaNs

`

Any suggestions ? Thanks a lot

-------------------------------------------------------------------------

2017-02-24T01:41:19Z rasbt

Interesting. I think it has nothing to do with tensorflow; my guess is that

[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)]

vs

[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)]

is the culprit!?

-------------------------------------------------------------------------

2017-02-24T01:45:30Z BerenLuthien

Thanks for response :) Any suggested solutions/to\_do\_list ?

Need use both

Tensorflow and

TSNE

in Jupyter notebook ....

BTW: just tried "from \_\_future\_\_ import division" in Python 2.x and did not solve the problem.

-------------------------------------------------------------------------

2017-02-24T02:05:30Z rasbt

Hm, not sure if that helps -- personally, I am not getting this mysterious issue anymore with

```

Python 3.5.3 |Continuum Analytics, Inc.| (default, Feb 22 2017, 20:51:01)

[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin

```

I am on Tf (now 1.0) as well, and I don't have this `Error: ValueError: array must not contain infs or NaNs` issue anymorewhen I execute

```

import numpy as np

from sklearn.manifold import TSNE

np.random.seed(1)

a = np.random.uniform(size=(100, 20))

TSNE(n\_components=2, random\_state=1).fit\_transform(a)

```

which previously didn't work.

Maybe try to create a new python 3.5 env and try the above-mentioned snippet to see if it works without error:

```

conda create -n yourenv python=3.5 numpy scipy scikit-learn

source activate yourenv

pip install tensorflow(-gpu)

```

-------------------------------------------------------------------------

2017-02-24T06:13:05Z BerenLuthien

Hi rasbt,

Yes I made TSNE work on Python 3.5.

However, for some other reason I'd better use Python 2.7, so I have to continue to explore ... cross fingers

Thanks for your help.

-------------------------------------------------------------------------

2017-02-24T06:29:22Z rasbt

Do you have an old(er) Miniconda/Anaconda 2.7 distro installed? In this case, maybe consider installing one of the more recent ones, or update your conda root or default python and give it another try (or create a new py 27 env by substituting the 3.5 by 2.7 in `conda create -n yourenv python=3.5 numpy scipy scikit-learn`) ? (not sure if this is really the reason, but I think `LLVM 4.2 (clang-425.0.28)` may be an issue; since the error doesn't seem to occur via `[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)]`)

-------------------------------------------------------------------------

2017-02-24T23:25:31Z BerenLuthien

Update: TSNE(perplexity=30, n\_components=2, init='pca', n\_iter=1000, \*\*method='exact'\*\*) make it worked ...

\*\*method='exact'\*\* was the trick.

-------------------------------------------------------------------------

2017-02-27T21:58:32Z bglick13

Also been having this problem. Using method='exact' seems to works for me, but it is so painfully slow. Is there really no other solution that people have found?

-------------------------------------------------------------------------

2017-02-28T10:03:07Z lesteve

Have you read https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-264029983 and https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-264087057 ?

The only way I managed to reproduce this problem was to install numpy with both pip and conda in the same conda environment. If you create a conda environment from scratch you should not have this problem.

In case your problem do not seem to match this description, please post the exact commands you ran to create your conda environment, so we can try to reproduce.

-------------------------------------------------------------------------

2017-05-11T21:51:45Z jsevo

Hi,

I read the above comments and can reproduce this. I re-ran code from a few weeks ago and now this issue appears. Here's a minimal example that now reproduces this issue:

```

from sklearn.manifold import TSNE

a = [[1,2,3],[4,5,6], [7,8,9]]

TSNE(n\_components=2,).fit\_transform(a)

```

And the output of

```

import platform; print(platform.platform())

import sys; print("Python", sys.version)

import numpy; print("NumPy", numpy.\_\_version\_\_)

import scipy; print("SciPy", scipy.\_\_version\_\_)

import sklearn; print("Scikit-Learn", sklearn.\_\_version\_\_)

```

is

```

Darwin-16.5.0-x86\_64-i386-64bit

Python 3.6.0 |Anaconda 4.3.0 (x86\_64)| (default, Dec 23 2016, 13:19:00)

[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)]

NumPy 1.12.1

SciPy 0.19.0

Scikit-Learn 0.18.1

```

Again, changing the method to exact (`TSNE(method='exact')`) gets rid of the error.

More generally, I have noticed wildly different results when using sklearn's TSNE (with identitical perplexity and other parameters) from the bh implementation published by Laurens van der Maaten and the MATLAB version. I wonder if there may be a connection?

-------------------------------------------------------------------------

2017-05-11T23:03:42Z glemaitre

Did you refer to https://github.com/scikit-learn/scikit-learn/issues/6665#issuecomment-264087057

-------------------------------------------------------------------------

2017-05-12T02:35:31Z jsevo

That fixed it. My apologies - I had separately uninstalled an reinstalled numpy, scikit learn and scipy, but not like in 6665.

-------------------------------------------------------------------------

2017-05-21T01:28:00Z OptimusCrime

I had the same problem as reported here, and I do not use conda.

My Python version is installed via brew on macOS Sierra 10.12.4

```

Python 3.6.1

scipy==0.19.0

scikit-learn==0.18.1

numpy==1.11.1

```

Adding `mode='exact'` solved my problem.

-------------------------------------------------------------------------

2017-06-01T17:13:03Z bbartoldson

@lesteve: i had this error using the setup you describe (two versions of numpy installed). simply updating the conda install of numpy to the same version as the pip install (1.12.1) did the trick for me. i did remove the pip numpy install, though, as i didn't intend to have two versions :)

-------------------------------------------------------------------------

2017-06-02T17:19:02Z walkon302

@lesteve: Thank you for the solution! I happened to have this error and then I found this discussion. Fix it right away after remove the duplicated version of numpy.

-------------------------------------------------------------------------

2017-08-15T15:31:17Z wolfiex

\*\*Replicated\*\* I have removed pip installs of numpy and updated conda.

Darwin-16.7.0-x86\_64-i386-64bit

('Python', '2.7.13 |Anaconda custom (x86\_64)| (default, Dec 20 2016, 23:05:08) \n[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)]')

('NumPy', '1.13.1')

('SciPy', '0.19.0')

('Scikit-Learn', '0.18.1')

It seems fine on my linux machine Linux:

Linux-3.0.101-0.47.71-default-x86\_64-with-SuSE-11-x86\_64

('Python', '2.7.12 |Anaconda 2.3.0 (64-bit)| (default, Jul 2 2016, 17:42:40) \n[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)]')

('NumPy', '1.12.1')

('SciPy', '0.19.1')

('Scikit-Learn', '0.18.1')

-------------------------------------------------------------------------

2017-08-15T16:56:44Z amueller

@wolfiex so you did

```

conda remove numpy --force -y

pip uninstall numpy -y

conda install numpy

```

Somewhat related I recommend you update to scikit-learn 0.19 which has some fixes in t-SNE

-------------------------------------------------------------------------

emove numpy --force -y

pip uninstall numpy -y

conda install numpy

```

Somewhat related I recommend you update to scikit-learn 0.19 which has some fixes in t-SNE

-------------------------------------------------------------------------