Description

Spectral embedding is an algorithm proposed by the following paper:

"Laplacian Eigenmaps for Dimensionality Reduction and Data Representation" M. Belkin, P. Niyogi, Neural Computation, June 2003; 15 (6):1373-1396

In page 6 of this paper, it defines

$$Lf = \lambda Df$$

where D is diagonal weight matrix, its entries are column sums of W, $D_{ii} = \sum_j W_{ji}$. $L = D - W$

Let $f_0, f_1, \ldots f_k - 1$ be the solutions of the above equations with $0 = \lambda_0 \leq \lambda_1 \leq \ldots \lambda_{k-1}$ **Note:** f_0 **is a constant vector**

Therefore, $(f_1, \ldots f_m)$ is lapacian eigenmaps

Let $g = D^{\frac{1}{2}} f$ and $L^{sym} = D^{\frac{-1}{2}}(D - W)D^{\frac{-1}{2}}$, we have

$$D^{\frac{-1}{2}}(D - W)D^{\frac{-1}{2}} * D^{\frac{1}{2}} f = \lambda D^{\frac{1}{2}} f$$

$$L^{sym} * g = \lambda g$$

In the code here:

```
laplacian, dd = graph_laplacian(adjacency,
          normed=norm_laplacian, return_diag=True)
```

`laplacian` is L^{sym} (with `norm_laplacian=True`), `dd` is $D^{\frac{1}{2}}$

In the code here

```
lambdas, diffusion_map = eigsh(laplacian, k=n_components,
        sigma=1.0, which='LM',
        tol=eigen_tol, v0=v0)
embedding = diffusion_map.T[n_components::-1] * dd
```

`diffusion_map` is g. and `embedding` is f.

However, $f = D^{\frac{-1}{2}} g$ which means

```
embedding = diffusion_map.T[n_components::-1] / dd
```

The same mistakes are in line 289 and line 313

Steps/Code to Reproduce

```
from sklearn.manifold.spectral_embedding_ import spectral_embedding
import numpy as np

matrix = np.zeros([4, 4])
matrix[0, 1] = 1
matrix[1, 0] = 1
matrix[0, 2] = 1
matrix[2, 0] = 1
matrix[1, 2] = 1
matrix[2, 1] = 1
matrix[2, 3] = 4
matrix[3, 2] = 4

f = spectral_embedding(matrix, n_components=1, drop_first=False)
```

Expected Results

Constant vector

```
array([[ 0.26726124],
       [ 0.26726124],
       [ 0.26726124],
       [ 0.26726124]])
```

Actual Results

```
array([[ 0.53452248],
       [ 0.53452248],
       [ 1.60356745],
       [ 1.06904497]])
```

Versions

```
>>> import platform; print(platform.platform())
Darwin-16.1.0-x86_64-i386-64bit
>>> import sys; print("Python", sys.version)
('Python', '2.7.9 (v2.7.9:648dcafa7e5f, Dec 10 2014, 10:10:46) \n[GCC 4.2.1 (Apple Inc. bui
>>> import numpy; print("NumPy", numpy.__version__)
('NumPy', '1.11.1')
>>> import scipy; print("SciPy", scipy.__version__)
('SciPy', '0.17.0')
>>> import sklearn; print("Scikit-Learn", sklearn.__version__)
('Scikit-Learn', '0.17.1')
```