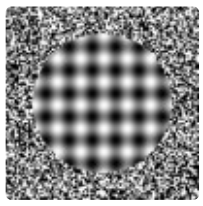

Anisotropic

Interactive Intro to Dimensionality Reduction

last run 2 days ago · Python notebook · 14633 vie...
using data from Digit Recognizer · Public

164 voters

Notebook

CAVEAT: Sorry but just note this notebook can be a bit slow to load probably due to the Plotly embeddings displaying a large number of points

Introduction

There already exists a plethora of notebooks discussing the merits of dimensionality reduction methods, in particular the Big 3 of PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis) and TSNE (T-Distributed Stochastic Neighbour Embedding). Quite a handful of these have compared one to the other but few have gathered all 3 in one go. Therefore this notebook will aim to provide an introductory exposition on these 3 methods as well as to portray their visualisations interactively and hopefully more intuitively via the Plotly visualisation library. The chapters are structured is as follows:

1. **Principal Component Analysis (PCA)** - Unsupervised, linear method
1. **Linear Discriminant Analysis (LDA)** - Supervised, linear method
1. **t-distributed Stochastic Neighbour Embedding (t-SNE)** - Nonlinear, probabilistic method

Lets go.

In [1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import seaborn as sns
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import matplotlib
%matplotlib inline

# Import the 3 dimensionality reduction methods
from sklearn.manifold import TSNE
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
```

Curse of Dimensionality & Dimensionality Reduction

The term "Curse of Dimensionality" has been oft been thrown about, especially when PCA, LDA and TSNE is thrown into the mix. This phrase refers to how our perfectly good and reliable Machine Learning methods may suddenly perform badly when we are dealing in a very high-dimensional space. But what exactly do all these 3 acronyms do? They are essentially transformation methods used for dimensionality reduction. Therefore, if we are able to project our data from a higher-dimensional space to a lower one while keeping most of the relevant information, that would make life a lot easier for our learning methods.

MNIST Dataset

For the purposes of this interactive guide, the MNIST (Mixed National Institute of Standards and Technology) computer vision digit dataset was chosen partly due to its simplicity and also surprisingly deep and informative research that can be done with the dataset. So let's load the training data and see what we have

In [2]:
```
train = pd.read_csv('../input/train.csv')
train.head()
```

Out[2]:

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel7