Student_3_Eigenvectors

April 14, 2021

1 Eigenvectors

In this notebook you can explore how the covariance matrix, data and eigenvectors are related. Feel free to play around with it!

[ ]:
```
%matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Ellipse
```

[ ]:
```
def error_ellipse(ax, xc, yc, cov, sigma=1, **kwargs):
    """
    https://github.com/megbedell/plot_tools/blob/master/error_ellipse.py
    Plot an error ellipse contour over your data.
    Inputs:
    ax : matplotlib Axes() object
    xc : x-coordinate of ellipse center
    yc : x-coordinate of ellipse center
    cov : covariance matrix
    sigma : # sigma to plot (default 1)
    additional kwargs passed to matplotlib.patches.Ellipse()
    """
    w, v = np.linalg.eigh(cov)  # assumes symmetric matrix
    order = w.argsort()[::-1]
    w, v = w[order], v[:, order]
    theta = np.degrees(np.arctan2(*v[:, 0][::-1]))  # * unpacks argument␣
↪instead of [0]
    ellipse = Ellipse(
        xy=(xc, yc),
        width=2.0 * sigma * np.sqrt(w[0]),
        height=2.0 * sigma * np.sqrt(w[1]),
        angle=theta,
        **kwargs
    )
    ellipse.set_facecolor("none")
    ax.add_artist(ellipse)

    return ax
```

[ ]:
```
n_population = 200
cov = np.array([[1, 0.5], [0.5, 1]])
mean = np.array([0, 0])
population = np.random.multivariate_normal(mean, cov, n_population)

w, v = np.linalg.eig(cov)
order = w.argsort()[::-1]
w, v = w[order], v[:, order]

fig, ax = plt.subplots(1, 1)

for ii in range(population.shape[0]):
    ax.plot(population[ii, 0], population[ii, 1], ".b")

ax = error_ellipse(
    ax, mean[0], mean[1], cov, ec="green", sigma=3, zorder=9999, label="3␣
 ↪$\sigma$"
)
ax = error_ellipse(
    ax, mean[0], mean[1], cov, ec="black", sigma=2, zorder=9999, label="2␣
 ↪$\sigma$"
)
ax = error_ellipse(
    ax, mean[0], mean[1], cov, ec="red", sigma=1, zorder=9999, label="1␣
 ↪$\sigma$"
)

# Eigenvalues
ax.arrow(
    mean[0],
    mean[1],
    np.sqrt(w[0]) * v[0, 0],
    np.sqrt(w[0]) * v[1, 0],
    width=0.1,
    color="red",
    length_includes_head=True,
)
ax.arrow(
    mean[0],
    mean[1],
    np.sqrt(w[1]) * v[0, 1],
    np.sqrt(w[1]) * v[1, 1],
    width=0.1,
    color="green",
    length_includes_head=True,
)
```

```
ax.set_aspect("equal", "box")
ax.set_xlim([-5, 5])
ax.set_ylim([-5, 5])
ax.set_xlabel("x")
ax.set_ylabel("y")

plt.show()
```

[ ]:

[ ]: