eigenvectors_notes

August 21, 2025

1 Eigenvectors — Concepts, Orthonormalization, and Diagonalization

This notebook focuses on eigenvectors, their computation and manipulation. It includes a Gram–Schmidt implementation to orthonormalize a basis, and demonstrates diagonalization when possible.

[1]:
```python
import numpy as np

def gram_schmidt(V):
    # V: matrix with vectors as columns
    n, k = V.shape
    Q = np.zeros((n, k))
    for i in range(k):
        v = V[:, i].copy()
        for j in range(i):
            v = v - np.dot(Q[:, j], V[:, i]) * Q[:, j]
        norm = np.linalg.norm(v)
        if norm < 1e-12:
            Q[:, i] = 0.0
        else:
            Q[:, i] = v / norm
    return Q

# Example: eigenvectors of symmetric matrix should be orthogonal
A = np.array([[5, 2, 0],
              [2, 3, 0],
              [0, 0, 4]], dtype=float)
vals, vecs = np.linalg.eig(A)
print('Eigenvectors (columns):\n', vecs)
Q = gram_schmidt(vecs)
print('\nAfter Gram-Schmidt (should match normalized eigenvectors up to sign):
 ↪\n', Q)
# Diagonalization check: A = V D V^{-1}
V = vecs
D = np.diag(vals)
V_inv = np.linalg.inv(V)
```

```
print('\nReconstruction error ||A - V D V^{-1}||_F:', np.linalg.norm(A - V @ D␣
↪@ V_inv))
```

```
Eigenvectors (columns):
 [[ 0.85065081 -0.52573111  0.        ]
 [ 0.52573111  0.85065081  0.        ]
 [ 0.          0.          1.        ]]

After Gram-Schmidt (should match normalized eigenvectors up to sign):
 [[ 0.85065081 -0.52573111  0.        ]
 [ 0.52573111  0.85065081  0.        ]
 [ 0.          0.          1.        ]]

Reconstruction error ||A - V D V^{-1}||_F: 7.021666937153402e-16
```

1.1 1. Notes

- Symmetric matrices have orthogonal eigenvectors; Gram–Schmidt isn't necessary but is useful for numeric stability.
- Not all matrices are diagonalizable; this is demonstrated by checking condition number of V.