PCA, RPCA, and ISOMAP

Outline

- PCA vs. Robust PCA
- Multi-Dimensional Scaling (MDS)

PCA vs. Robust PCA

Goal.

- Reduce the dimensions of the feature matrix $X \in \mathbb{R}^{n \times d}$
- Compute $Z \in \mathbb{R}^{n \times k}$ where $k << d$
- k is user-defined

Objective function

$$\begin{aligned} f(W, Z) &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{d} (w_j^T z_i - x_{ij})^2 \\ &= \frac{1}{2} ||ZW - X||_F^2 \end{aligned} \tag{1}$$

PCA vs. Robust PCA

$$f(W,Z) = \frac{1}{2}||ZW - X||_F^2 \quad (Z \in \mathbb{R}^{n\times k}, W \in \mathbb{R}^{k\times d}) \tag{2}$$

Algorithm.

1. Optimize w.r.t Z while keeping W fixed

$$Z^* = \arg\min_Z \frac{1}{2}||ZW - X||_F^2$$

2. Optimize w.r.t W while keeping Z fixed

$$W^* = \arg\min_W \frac{1}{2}||ZW - X||_F^2$$

3. Repeat steps 1 and 2 until convergence

PCA vs. Robust PCA

At tain time: * Get Z * Train a logistic regression

- Motivation: PCA is used for dimensionality reduction (to speed up training and testing time)

PCA

$$\arg\min_{Z,W} \frac{1}{2}||ZW - X||_F^2 \quad (Z \in \mathbb{R}^{n\times k}, W \in \mathbb{R}^{k\times d}) \qquad (3)$$

Given code. For optimizing w.r.t Z and W, respectively.

```
def _fun_obj_z(self, z, w, X, k):
    n,d = X.shape
    Z = z.reshape(n,k)
    W = w.reshape(k,d)
    R = np.dot(Z,W) - X
    f = np.sum(R**2)/2
    g = np.dot(R, W.transpose())
    return f, g.flatten()

def _fun_obj_w(self, w, z, X, k):
    n,d = X.shape
    Z = z.reshape(n,k)
    W = w.reshape(k,d)
    R = np.dot(Z,W) - X
    f = np.sum(R**2)/2
    g = np.dot(Z.transpose(), R)
    return f, g.flatten()
```

Robust PCA

$$\arg\min_{Z,W} ||ZW - X||_1 \qquad (Z \in \mathbb{R}^{n\times k}, W \in \mathbb{R}^{k\times d}) \tag{4}$$

Use the smooth approximation of the L1 norm

$$\alpha \approx \sqrt{\alpha^2 + \epsilon}$$

Write the code for optimizing w.r.t Z and W, respectively.

```
def _fun_obj_z(self, z, w, X, k):
    f = ?
    g = ?
    return f, g.flatten()

def _fun_obj_w(self, w, z, X, k):
    f = ?
    g = ?
    return f, g.flatten()
```

Multi-Dimensional Scaling

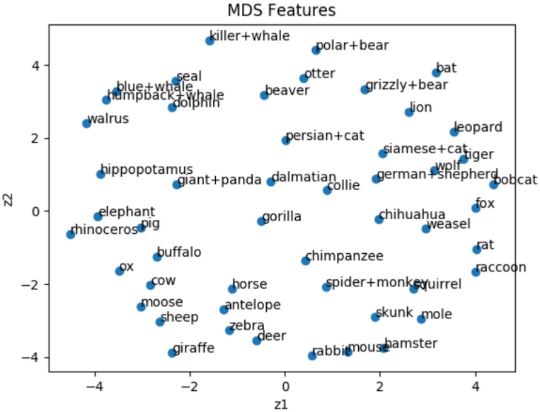
Goal To represent the data points in lower dimensional space $k << d$ such that the similarities between them is retained.

$$f(Z) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (||z_i - z_j|| - ||x_i - x_j||)^2 \quad (z_i \in \mathbb{R}^k, x_i \in \mathbb{R}^d) \tag{5}$$

- Optimize for the new data points directly
  - Is MDS parametric or not ?

Multi-Dimensional Scaling

$$f(Z) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (||z_i - z_j|| - ||x_i - x_j||)^2 \quad (z_i \in \mathbb{R}^2, x_i \in \mathbb{R}^d) \tag{6}$$



Multi-Dimensional Scaling (Code)

$$\arg\min_{Z} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(||z_i - z_j|| - ||x_i - x_j||)^2 \qquad (z_i \in \mathbb{R}^k, x_i \in \mathbb{R}^d)$$

```
def compress(self, X):
    n = X.shape[0]
    k = self.k

    # Compute Euclidean distances (Unique to MDS)
    D = utils.euclidean_dist_squared(X,X)
    D = np.sqrt(D)

    # Initialize low-dimensional representation with PCA
    Z = PCA(k).fit(X).compress(X)

    # Solve for the minimizer
    z = find_min(self._fun_obj_z, Z.flatten(), 500, False, D)
    Z = z.reshape(n, k)
    return Z
```

Assignment: Implement ISOMAP

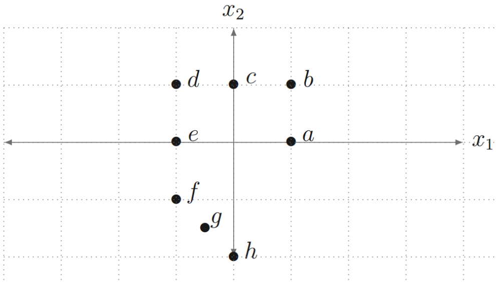
Same objective function as MDS:

$$\arg\min_{Z} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (||z_i - z_j|| - ||x_i - x_j||)^2 \qquad (z_i \in \mathbb{R}^k, x_i \in \mathbb{R}^d)$$

```
def compress(self, X):
    n = X.shape[0]
    k = self.k

    # Compute Euclidean distances (Unique to MDS)

    YOUR CODE HERE. DISTANCES ARE COMPUTED USING GEODISTIC DISTANCE

    # Initialize low-dimensional representation with PCA
    Z = PCA(k).fit(X).compress(X)

    # Solve for the minimizer
    z = find_min(self._fun_obj_z, Z.flatten(), 500, False, D)
    Z = z.reshape(n, k)
    return Z
```

ISOMAP: Compute Geodistic distance

- Create a distance matrix using geodesic distances, where k



$= 2$

ISOMAP: Answer

	a	b	c	d	e	f	g	h
a	0	1	$\sqrt{2}$	$\sqrt{2}+1$	$\sqrt{2}+2$	$\sqrt{2}+3$	$1.5\sqrt{2}+3$	$2\sqrt{2}+3$
b		0	1	2	3	4	$0.5\sqrt{2}+4$	$\sqrt{2}+4$
c			0	1	2	3	$0.5\sqrt{2}+3$	$\sqrt{2}+3$
d				0	1	2	$0.5\sqrt{2}+2$	$\sqrt{2}+2$
e					0	1	$0.5\sqrt{2}+1$	$\sqrt{2}+1$
f						0	$0.5\sqrt{2}$	$\sqrt{2}$
g							0	$0.5\sqrt{2}$
h								0