
Machine Learning HW7

MLTAs

ntumlta2019@gmail.com

Outline

- Task Description - Unsupervised Learning
 - Task A: PCA of colored faces
 - Task B: Image clustering
- Kaggle
- Requirements & Regulation
- Grading Policy
- FAQ

Outline

- Task Description - Unsupervised Learning
 - Task A: PCA of colored faces
 - Task B: Image clustering
- Kaggle
- Requirements & Regulation
- Grading Policy
- FAQ

PCA of colored faces - outline ^{1/3}

- 學習用 numpy 實做 PCA 以達到 dimensionality reduction 的目的
- 數據集來自 Aberdeen University 的 Prof. Ian Craw, 並經過助教們的挑選及對齊, 總共有 415 張 600 X 600 X 3 的彩圖。
- 連結: [Link](#) (5/9 已更新 !)
- Usage: tar zxvf Aberdeen.tar.gz

PCA of colored faces - requirements 2/3

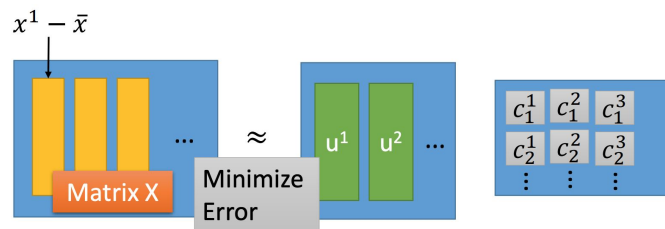
1. 只能用 [scikit-image](#) 讀寫圖片
2. 只能用 [numpy.linalg.svd](#) 或 [np.linalg.eig](#) 實做PCA
3. 將各張圖片分解成Eigenvector (Eigenface)

$$x - \bar{x} \approx c_1 u^1 + c_2 u^2 + \dots + c_K u^K = \hat{x}$$

Reconstruction error:

$$\| (x - \bar{x}) - \hat{x} \|_2$$

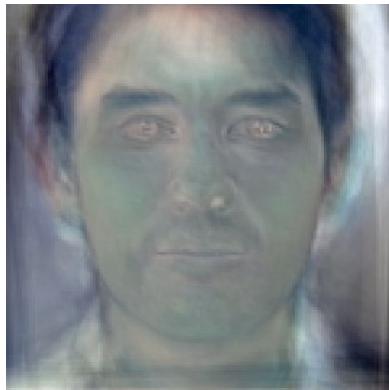
Find $\{u^1, \dots, u^K\}$ minimizing the error



4. 每一個Eigenface 都對應到不同的Eigenevalue
5. 將Eigenface由大到小排列, 選出前五大的Eigenface 重建出圖片

PCA of colored faces - reminder ^{3/3}

- 請記得先減去平均再計算 Eigenfaces, Eigenvalues
- Eigenfaces 是奇怪的顏色是正常的, 如右上圖(第十個 eigenface)
- 因為 Eigenfaces 會有負值, 因此在畫圖時, 請用以下方式轉換:
 - `M -= np.min(M)`
 - `M /= np.max(M)`
 - `M = (M * 255).astype(np.uint8)`
- 程式只會執行最多七分鐘。
- 只能 import [numpy](#) 和 [skimage](#) (and other python standard library)
- 程式的結果是有標準答案的(可容許每個值相差 ± 3 以內), 可以事先和同學比看看
- 可以參考老師的投影片:[Link](#)



Outline

- Task Description - Unsupervised Learning
 - Task A: PCA of colored faces
 - Task B: Image clustering
- Kaggle
- Requirements & Regulation
- Grading Policy
- FAQ

Image clustering - outline ^{1/7}

- 目標: 分辨給定的兩張 images 是否來自同一個 dataset
 - 所有的 image 都來自兩個不同的 class (CelebA & cifar10)
 - 除了 image 都是 $32 \times 32 \times 3$ 的圖片, 沒有任何 label
 - 只能用我們給的 data, 不能使用額外的 dataset, 也不能使用額外資料 train 的 model

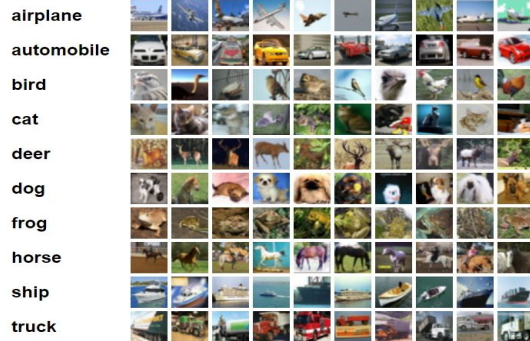


Image clustering - data 2/7

- images.zip
 - Usage: unzip images.zip
 - 會得到一個叫images/ 的資料夾, 裡面總共有 40000 張 RGB圖片, 大小都是32*32*3
- visualization.npy
 - 在 kaggle deadline 之後會公布一個小型的 dataset
 - 包含 5000 張 images, 前 2500 張 images 跟後 2500 張 images 是分別從兩個 dataset 得到的
 - 請用訓練的模型在report對這個 dataset 做 visualization

Image clustering - data 3/7

- test_case.csv
 - 每一行都有 id, image1_name, image2_name, 總共有 1,000,000 筆測資
 - id: test case index
 - image1_name: 對應到 images/ 裡的圖片的名稱
 - image2_name: 對應到 images/ 裡的圖片的名稱
- sample_submission.csv
 - 第一行是 "id, label" (5/11 update)
 - 之後每一行都會有 test case ID, 以及對這個 test case 的 prediction
 - 如果 test case 的兩張 image 預測後是來自同一 dataset, Ans 的地方就是 1, 反之是 0

Image clustering - methods 4/7

- 如果直接在原本的 image 上做 cluster, 結果會很差 (有很多冗餘資訊)

=> 需要更好的方式來表示原本的image

- 為了找出這個更好的方式, 可以先將原始 image 做 dimension reduction, 用比較少的維度來描述一張 image

Image clustering - requirements 5/7

1. 請實作用 **autoencoder** 將40000張圖片降維
2. 再利用降維過的latent code做分類
3. 預測1000000筆測資是否來自相同的dataset

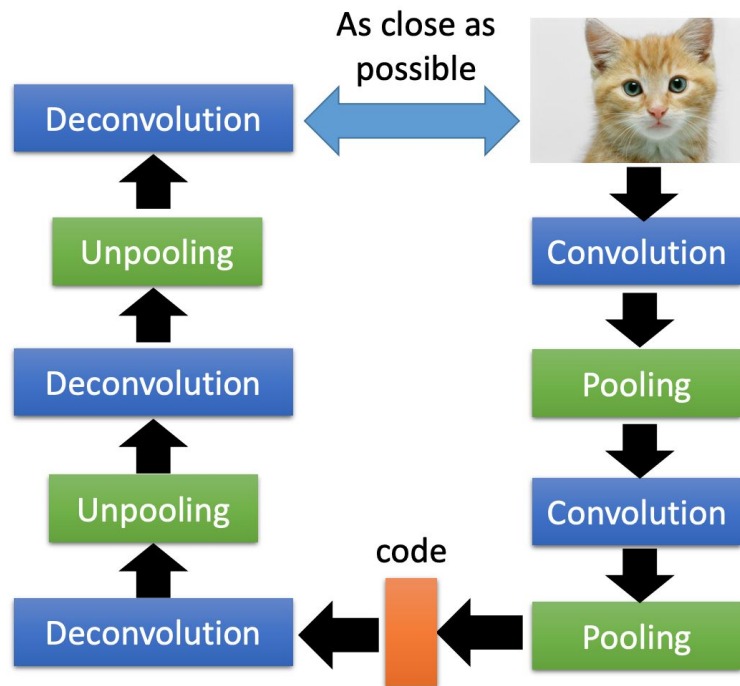


Image clustering - methods (cont.) 6/7

- 接著對降維過後過後的數據做 cluster
 - cluster: 可以試試 K-means
- 或者你可以衡量兩個降維過後的 images, 他們之間的相似度 (similarity)。如果相似度大於一個設定好的 threshold, 就把這兩個 images 當成同一類別
 - 算 similarity 的方法: euclidean distance, cosine similarity.....

Image clustering - methods (cont.) 7/7

- 其他可能有幫助的事：
 - 必須找個方法來衡量方法的好壞，一個直覺的方法是利用降維過後的feature 去 reconstruct 成原本的 image。如果 reconstruct 的結果越接近原本的 image，可以一定程度的代表你抽出來的 feature 越好
 - 對原始 image 做 data augmentation
 - try different number of cluster
 - 看看老師 unsupervised learning 上課內容

Outline

- Task Description - Unsupervised Learning
 - Task A: PCA of colored faces
 - Task B: Image clustering
- Kaggle
- Requirements & Regulation
- Grading Policy
- FAQ

Kaggle - Info ^{1/2}

- Kaggle 連結 : <https://www.kaggle.com/c/ml2019spring-hw7>
 - 個人進行, 不需組隊
 - 隊名:
 - 修課學生: 學號_任意名稱 (ex: b08901777_活大好好吃)
 - 旁聽: 旁聽__任意名稱
 - 每天上傳上限 5 次
 - Leaderboard上所顯示為public score, 在Kaggle Deadline前可以選擇2份submission作為private score的評分依據。
 - test set的資料將被分為兩份, 一半為public, 另一半為private。
 - 最後的計分排名將以2筆自行選擇的結果, 測試在private set上的準確率。
- ★ kaggle名稱錯誤者的分數將x0.7。

Kaggle - format 2/2

- 預測 1000000 筆 testing data 是否來自相同的dataset, 將預測結果上傳至kaggle
 - Upload format : csv file
 - 第一行必須是 id,label
 - 第二行開始, 每行分別為id值及預測結果 (binary), 以逗號隔開
 - 預測後是來自同一 dataset, label 的地方就是 1, 反之是 0
 - Evaluation: Accuracy
- 範例格式如右

```
sample_submission.csv x
1 id,label
2 0,0
3 1,0
4 2,0
5 3,0
6 4,0
7 5,0
8 6,0
9 7,0
10 8,0
11 9,0
12 10,0
13 11,0
14 12,0
15 13,0
16 14,0
17 15,0
18 16,0
19 17,0
20 18,0
21 19,0
22 20,0
```

Outline

- Task Description - Unsupervised Learning
 - Task A: PCA of colored faces
 - Task B: Image clustering
- Kaggle
- Requirements & Regulation
- Grading Policy
- FAQ

Requirements

- Task A: PCA of colored faces
 - 不用上傳kaggle
 - 用PCA實作出 eigenface, 及eigenface reconstruct 的結果
 - 回答report 問題
- Task B: Image clustering
 - 將預測結果上傳kaggle
 - 用autoencoder 實作降維
 - 回答report問題
 - 不能使用額外的data訓練, 也不能使用pre-trained model
 - 不能 call 其他線上 API

Regulation 1/3

- **Python Only**, 請使用 Python 3.6
- Python standard library are available
- **PCA of colored faces:**
 - Numpy ≥ 1.14
 - scikit-image == 0.15.0
- **Image clustering:**
 - Numpy ≥ 1.14 , Pandas ≥ 0.20
 - Keras == 2.2.4, Tensorflow $\geq 1.12.0$, pytorch == 1.0.1
 - Scipy == 1.2.1
 - scikit-image == 0.15.0
 - scikit-learn == 0.20.3
 - Pillow == 6.0.0
- **若需要其它套件, 請及早來信詢問。** 若 import 有發生錯誤, 分數將x0.7



Regulation - GitHub 2/3

- 你的 github 上 ML2019SPRING/hw7/ 中請包含：
 - report.pdf
 - pca.sh (for PCA of colored face 那題) (限時7分鐘)
 - cluster.sh (for image clustering 那題, 限制用autoencoder實作) (限時10分鐘)
 - your python files
 - your model files (can be loaded by your python file)
- 請不要上傳 dataset, 請不要上傳 dataset, 請不要上傳 dataset.
- 如果你的 model 超過 github 的最大容量, 可以考慮把 model 放在其他地方 (<http://slides.com/sunprinces/deck-16#/2%EF%BC%89>)。
- model 可以是多個檔案, 例如 keras model。如果你的 code 需要極長的執行時間, 可以把 image cluster 後的結果寫進一個 file, 並在執行時讀取它。

Regulation - Script Usage ^{3/3}

- 以下的路徑，助教在跑的時候會另外指定，請保留可更改的彈性，不要寫死

a. PCA of colored faces:

`bash pca.sh <images path> <input image> <reconstruct image>`

e.g. `bash pca.sh Aberdeen/ 87.jpg 87_reconstruct.jpg`

b. Image clustering:

`bash cluster.sh <images path> <test_case.csv path> <prediction file path>`

e.g. `bash cluster.sh images/ test_case.csv ans.csv`

- Script 所使用之模型，如 hdf5, pt, pickle 檔等，可以於程式內寫死路徑，助教會 cd 進 hw7 資料夾執行 reproduce 程序。

Outline

- Task Description - Unsupervised Learning
 - Task A: PCA of colored faces
 - Task B: Image clustering
- Kaggle
- Requirements & Regulation
- Grading Policy
- FAQ

Grading Policy - Deadline ^{1/7}

- Early Simple Deadline: 2019/05/16 11:59:59 (GMT+8)
- Kaggle Deadline: 2019/05/23 11:59:59 (GMT+8)
- Github Deadline: 2019/05/24 23:59:59 (GMT+8)

助教會在deadline一到就clone所有程式, 並且**不再重新clone任何檔案**

若遲交請填寫遲交表單: [Link](#)

Grading Policy - Evaluation (5% + Bonus 1%) ^{2/7}

- (1%) 超過public leaderboard的simple baseline分數
- (1%) 超過public leaderboard的strong baseline分數
- (1%) 超過private leaderboard的simple baseline分數
- (1%) 超過private leaderboard的strong baseline分數
- (1%) 2019/05/16 11:59:59 (GMT+8)前超過public simple baseline
- (BONUS 1%) private leaderboard 排名前五名且於助教時間上台分享的同學

Grading Policy - Report _{3/7}

1. PCA of color faces:

- a. (0.5%) 請畫出所有臉的平均。
- b. (0.5 %) 請畫出前五個 Eigenfaces, 也就是對應到前五大 Eigenvalues 的 Eigenvectors。
- c. (0.5%) 請從數據集中挑出任意五張圖片, 並用前五大 Eigenfaces 進行 reconstruction, 並畫出結果。
- d. (0.5%) 請寫出前五大 Eigenfaces 各自所佔的比重, 也就是 $\frac{s_i}{\sum s_j}$ 請用百分比表示並四捨五入到小數點後一位。

Grading Policy - Report 4/7

2. Image clustering:

- a. (1%) 請實作兩種不同的方法，並比較其結果(reconstruction loss, accuracy)。(不同的降維方法或不同的 cluster 方法都可以算是不同的方法)
- b. (1%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。(用 PCA, t-SNE 等工具把你抽出來的 feature 投影到二維，或簡單的取 feature 的前兩維)
其中visualization.npy 中前 2500 個 images 來自 dataset A, 後 2500 個 images 來自 dataset B, 比較和自己預測的 label 之間有何不同。

★ visualization.npy 會在kaggle請用train好的模型去預測

Grading Policy - Report 5/7

2. Image clustering:

- c. (1%) 請介紹你的model架構(encoder, decoder, loss function...), 並選出任意32張圖片, 比較原圖片以及用decoder reconstruct的結果。



★ 請勿使用此範例圖片！！

Grading Policy - Report 6/7

- 限制
 - 檔名必須為 report.pdf !!!
 - 檔名必須為 report.pdf !!!
 - 檔名必須為 report.pdf !!!
 - 請用中文撰寫 report (非中文母語者可用英文)
 - 保留各題標題
 - 請標明系級、學號、姓名，並按照report模板回答問題，切勿隨意更動題號順序
 - 若有和其他修課同學討論，請務必於題號前標明collaborator (含姓名、學號)
- Report模板連結
 - 連結: [Link](#)
- 截止日期同 GitHub Deadline: **2019/05/24 23:59:59 (GMT+8)**

Grading Policy - Other Policy 7/7

- **Lateness**

- Github 遲交一天(不足一天以一天計算) hw7 所得總分將 $\times 0.7$
- **不接受程式 or 報告單獨遲交**
- 不足一天以一天計算, 不得遲交超過兩天, 有特殊原因請找助教。
- Github 遲交表單: 遲交請先上傳遲交檔案至自己的github 後再填寫遲交表單, 助教群會以表單填寫時間作為繳交時間手動clone 檔案。

- **Script Error**

- 當 **script 格式錯誤**, 造成助教無法順利執行, 請在公告時間內寄信向助教說明, 修好之後重新執行所得 kaggle 部分分數將 $\times 0.7$ 。
- 可以更改的部分僅限syntax及io的部分, 不得改程式邏輯或是演算法, 至於其他部分由助教認定為主。
- 不接受任何 py 檔的 coding 錯誤更改

FAQ

- 若有其他問題，請寄信至助教信箱，**請勿直接私訊助教**。
- 有問題建議可以在 FB Group 裡面留言發問，可能很多人都有一樣的問題
- 助教信箱 ntumlta2019@gmail.com

