**CS302 Introduction to machine learning**



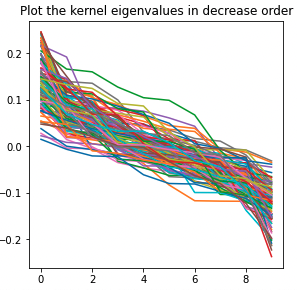
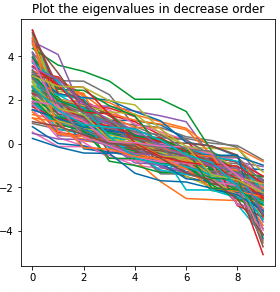
School of Undergraduate Studies, DGIST

201611161 정원균

Homework report 4.

2022. 06. 07

**Q1. Run the PCA and kernel PCA functions on the Digit-2-Space (100 training images used in Assignment 3). Plot the mean image and the first 10 eigenvectors (as images). Plot the eigenvalues (in decreasing order) as a function of dimension. Describe what you find in both plots.**

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

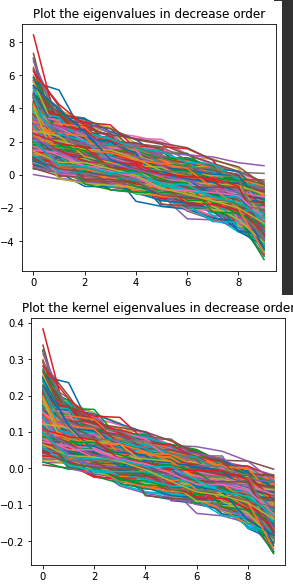
자동 생성된 설명

과제 3번에서 사용하였던 Mnist 손글씨 데이터에서 각 클래스 별로 100개씩 random sampling한 총 1000개의 데이터를 학습에 사용하였다. 1000개의 학습데이터를 10개의 component로 PCA와 ‘rbf’ Kernel PCA을 하였다. 숫자 2에 해당하는 Mean image와 10개의 eigen vector 로 Dimension Reduction을 한 다음에 복원한 이미지를 보여준다. Mean image는 2-digit space에 있는 2에 해당하는 모든 image들이 다 합쳐진 값의 평균이미지를 보여주기 때문에 두껍고 퍼진 2를 나타낸다. 또 10개의 eigen vector로 차원을 줄이고 복원한 이미지는 10개의 feature들만 가지고 표현해야하기 때문에 조금 불분명한 이미지가 나오지만 충분히 2를 표현함을 알 수 있다. PCA와 Kernel PCA을 통해서 얻은 Eigen value들의 plot을 그려보면 대체적으로 유사한 형태를 보인다는 것을 알 수 있다.

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명**Q2. Run the PCA and kernel PCA functions on all 1000 training images used in Assignment 3. Plot the mean image and the first 10 eigenvectors (as images). Plot the eigenvalues (in decreasing order) as a function of dimension. Describe what you find in both plots. Compare these plots to the ones you created in (1).**

텍스트이(가) 표시된 사진

자동 생성된 설명

역시 1번에서 Sampling한 100개씩 10개 class에 대해서 PCA와 kernel PCA을 적용하였다. Mean image같은 경우에 0부터 9까지 모든 이미지들의 평균값을 나타내기 위해서 0,3,9등등 다양한 숫자들이 섞여서 보임을 알 수 있다. 하지만 이를 10개 eigen vector로 Reduction을 한 다음 복원한 결과 0과 비슷한 이미지를 보인다. 이는 여러 이미지가 섞인 이미지에서 가장 weight가 큰 주 성분만을 뽑아서 차원 축소를 한 뒤 다시 복원하기 때문에 mean image에서 보이는 숫자의 태두리 모양이 적용이 크게 되어서 이 부분이 복원 시에 뚜렷하게 나타났다. 따라서 Q1의 경우엔 실제 2데이터를 복원한 이미지라고 할 수 있지만 Q2에서는 복원한 이미지가 0과 비슷하지만 실제로 0이미지를 차원 축소 한 후 복원했다고 단정지을 수는 없다. 역시 PCA와 Kernel PCA를 통해서 구해진 eigen vector을 plot으로 나타내면 다음과 같다. 또한 PCA를 적용시킨 data수가 더 많기 때문에 Q1. 에서 보이는 plot보다 더 빽빽한 plot이 나온다는 것을 알 수 있고 PCA와 kernel PCA가 비슷한 형태를 보인다는 것 역시 알 수 있다.

**Q3. Run K-means clustering on the reduced features using the PCA and kernel PCA, respectively. Compute “Rand index” and “mutual information-based score” on the training data. Explain your findings.**

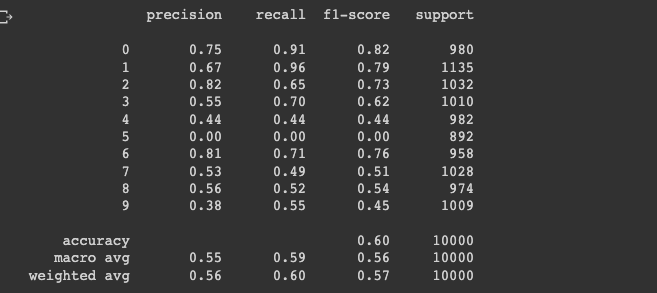
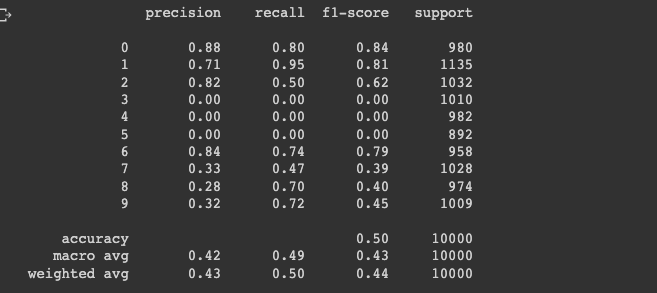
텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

오른쪽 코드는 PCA(10)을 적용시킨 후 Prediction을 진행한 값과 실제 값에 대해서 rand\_score와 mutual\_info\_score을 계산하였다. 그 결과 0.869의 rand\_score를 보였고 normalized\_mutual\_info\_score는 0.492가 나왔다. 주어진 수치만 놓고 보았을 때 K-means clustering가 87%이상의 정확도를 보이며 각 클러스터를 비교할 수 있을 정도의 상호 의존도를 보임을 알 수 있다. Kernel PCA을 적용한 결과를 보면 그 결과 0.873의 rand\_score를 보였고 normalized\_mutual\_info\_score는 0.498가 나왔다. 그냥 PCA를 돌릴 때 보다 근소하게 높은 rand\_score값을 내었고 상호의존도 역시 0.5 미만의 점수를 기록하여 각 클러스터를 구분할 정도의 정보를 낼 수 있다는 것을 알 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명****텍스트이(가) 표시된 사진

자동 생성된 설명**Q4. Classify the MNIST test data set using 1-NN classifier like Assignment 3. Compute accuracy scores and compare them with the accuracy in Assignment 3.**

텍스트이(가) 표시된 사진

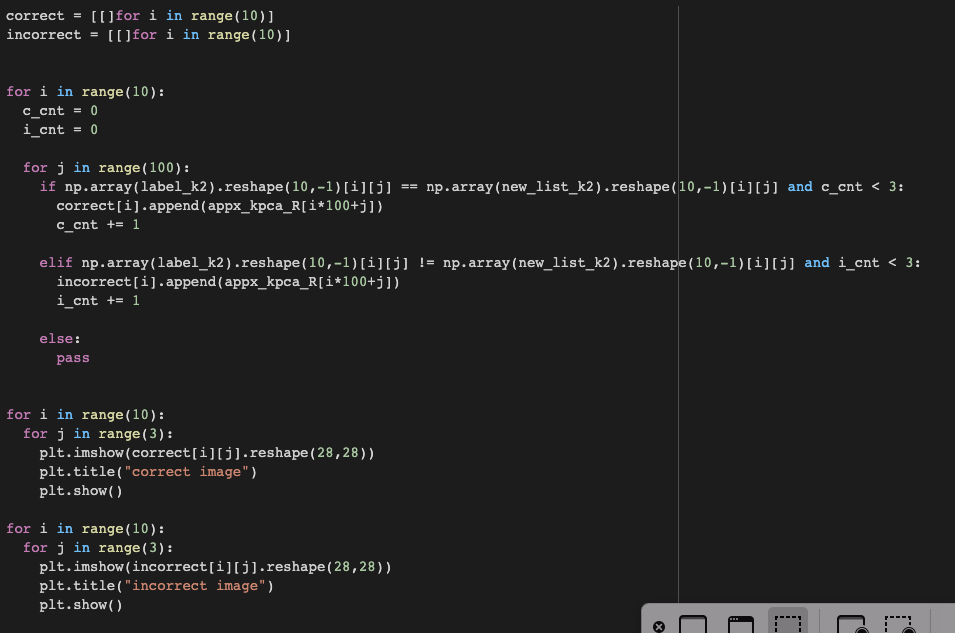
자동 생성된 설명

Accuracy from HW3

다음은 과제 3번에서 했던 방식과 유사하게 Cluster의 center을 구한 다음에 이를 기준으로 1-NN classifier fitting을 하고 fitting된 classifier을 실제 Mnist data set X\_test와 Y\_test에 적용하여서 실제로 PCA와 Kernel PCA을 사용한 분류 모델이 테스트 데이터에 대해서도 분류를 잘 할 수 있는지 측정하였다. 그 결과 그냥 PCA를 썼을 경우에는 50%정도의 정확도를 보임을 알 수 있고 Kernel PCA를 사용하였을 때는 이보다 약 10%정도 높은 60%의 정확도를 보임을 알 수 있다. Kernel PCA를 사용하면 non-linear한 decision boundary를 표현하는 등 그냥 PCA를 적용할 때 보다 조금 더 정교한 모델을 제공하기 때문에 성능적으로 일반 PCA보다 좋다고 분석할 수 있다. 또한 과제 3번에서 PCA를 적용하지 않고 kmeans clustering과 1-nn classifier을 적용해서 test set에 적용한 45%의 정확도 보다 각각 약 5%, 15% 정도의 성능이 향상되었음을 알 수 있다. 이로 미루어 보았을 때 Mnist data set 분류할 때 PCA 기법을 적용하면 그냥 Clustering을 했을 때 보다 조금 더 좋은 정확도를 보여준다는 것을 알 수 있다.

도시이(가) 표시된 사진

자동 생성된 설명도시이(가) 표시된 사진

자동 생성된 설명**Q5. Visualize 3 correctly classified and 3 incorrectly classified images for each class. Explain your findings.**

다음은 PCA를 통해서 복원한 image를 대상으로 제대로 Classification을 하는 경우엔 correct data set으로, 제대로 Classification을 하지 못하는 이미지는 incorrect data set으로 분류해서 각 class별로 3개씩 출력한 결과이다. 실제로 correct 이미지의 경우에 비교적 clear하게 숫자를 구분할 수 있는 형태를 보이지만 incorrect 이미지의 경우엔 숫자가 뭉개져서 나오거나 두 숫자 사이에서 구별하기 애매한 모양을 보임을 알 수 있다. (예를 들어 1을 예측하는데 3인지 1인지 확실하게 구분할 수 없는 이미지가 나옴) 실제로 prediction을 진행하여서 target label을 출력해보면 우리가 원하는 대로 1부터 9까지의 classification 결과가 나오기를 바라지만 실제로 중복된 label값이 나와서 다 깔끔하게 나누지는 못 하였다. 가장 대표적인 예시로 4에 대한 classification을 들 수 있는데 correct 데이터에서도 보다시피 출력된 이미지가 4인지 9인지 명확하게는 구분되지 않음을 알 수 있다. 따라서 이러한 경우에는 10개의 축소된 feature들로만 복원하지 않고 주성분의 수를 늘리거나 training에 사용하는 data set을 100개씩만 활용하는 것이 아닌 더 많은 data set을 이용한다면 조금 더 명확한 이미지가 나올 것이며 PCA와 KPCA의 정확도도 4번에서 보았던 정확도 보다 더 높은 값을 보일 것이라고 예측할 수 있다.