

머신러닝 hw5 – LDA 적용하기

1. 개요

- Mnist data 에 LDA 를 적용하여 차원을 축소하고, kNN 과 randomforest classifier 를 이용한 분류 정확도가 얼마나 나오는지 계산하였다. 이 결과를 eigenspace 에 project 한 데이터의 분류 정확도와 비교하였다.

2. LDA 알고리즘 설명

- 데이터의 각 레이블에 따라 10 개 클래스로 분류하였다.
- 데이터 전체의 평균 벡터와 각 클래스의 평균 벡터를 이용하여 within-class scatter matrix 와 between-class scatter matrix 를 구했다.
- $S_w^{-1}S_B w = \lambda w$ 공식을 이용하여 $S_w^{-1}S_B$ 의 eigenvector 를 구하는 방식으로 w 를 구하였다.
 - > S_w 는 singular matrix 이므로 역행렬을 구하기 위해 pseudo-inverse 를 사용하였다.
- eigenvalue 가 큰 순으로 2,3,5,9 개의 eigenvector 를 이용해 matrix 를 만든다.
- 각 matrix 에 원본 데이터를 곱해, 2,3,5,9 차원으로 project 된 데이터를 얻는다.

3. 결과

	kNN	randomforest (estimator=10)
raw data	k:1, score: 91.08	score: 94.60
	k:5, score: 90.60	
	k:10, score: 89.54	
eigenspace (Dim=2)	k:1, score: 36.62	score: 39.50
	k:5, score: 39.90	
	k:10, score: 41.24	
eigenspace (Dim=3)	k:1, score: 41.76	score: 45.78
	k:5, score: 46.04	
	k:10, score: 47.68	
eigenspace (Dim=5)	k:1, score: 63.90	score: 66.96
	k:5, score: 68.48	
	k:10, score: 69.70	
eigenspace (Dim=9)	k:1, score: 83.40	score: 79.84
	k:5, score: 85.66	
	k:10, score: 85.72	
LDA (Dim=2)	k:1, score: 44.72	score: 48.30
	k:5, score: 48.32	
	k:10, score: 49.66	
LDA (Dim=3)	k:1, score: 61.26	score: 67.58
	k:5, score: 66.24	
	k:10, score: 66.74	
LDA (Dim=5)	k:1, score: 73	score: 77.76
	k:5, score: 76.94	
	k:10, score: 76.64	
LDA (Dim=9)	k:1, score: 78.08	score: 84.12
	k:5, score: 80.22	
	k:10, score: 80.06	

LDA 와 PCA 를 적용한 뒤 각각 사영된 데이터에 대해 kNN, randomforest 분류를 진행한 결과이다.