Assignment #4 Report

컴퓨터교육과 / 2020310050 / 김보민

Intro

해당 프로그램은 이미지 1000개에 대해서 추출한 SIFT features 에 대하여 VLAD(Vector of Locally Aggregated Descriptors)를 사용해 discriptor를 생성한다.

과제 수행에 앞서 VLAD에서 분류에 사용되는 K-Means 알고리즘의 cluster 개수에 대한 고민이 많았었다. descriptor에서 최대 1024차원을 가질 수 있다는 과제 조건에서 힌트를 얻어 8개의 cluster로 확정할 수 있었다($1024 = 128 \times 8$).

Process

SIFT 파일에서 feature를 불러와 이미지에 따른 구별없이 배열에 저장한다.

```
6   data=[]
7   for num in range(N_IMG):
8       features=get_SIFT_features(str(num).zfill(5))
9       data.extend(features)
10  data=np.array(data)
```

총 702,120개의 features에서 100,000개를 무작위로 추출하여 초기 cluster 배열을 생성한다. 이 과정에서 처음에는 k-means을 적용했으나, 이후 성능 향상을 위해 k-means++을 사용하였다.

```
rand10e4=get_rand_array(data,N_10e4)
clustersInit=np.array(init_kmeans_plus(rand10e4)) #k-means++
# clusters=get_rand_array(rand10e4,N_CLUSTERS) #k-means
```

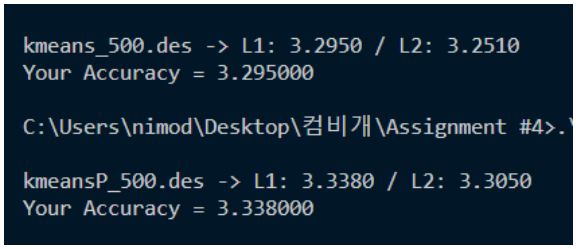
```
def init_kmeans_plus(data):
    clusters=[random.choice(data)]
    for _ in range(N_CLUSTERS-1):
        distance=np.array([min([L2_distance(feature,cluster) for cluster in clusters]) for feature in data])
        cluster=data[np.argmax(distance)]
        clusters.append(cluster)
    return clusters
```

cluster 계산 후 descriptor vector를 생성한다.

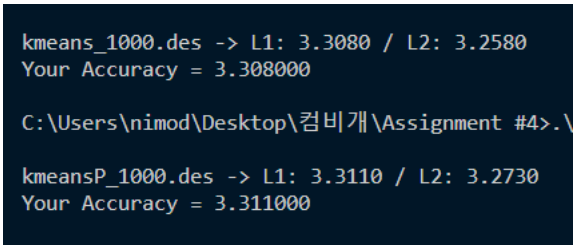
```
clustersFinal=cluster_centers_kmeans(rand10e4,clustersInit)

descriptor = np.zeros((N_IMG, MAX_DIM))
for num in range(N_IMG):
    features=get_SIFT_features(str(num).zfill(5))
    descriptor[num]=get_descriptor(features,clustersFinal)
```

Result

K-means과 K-means++ 알고리즘에 대한 결과는 다음과 같다. seed가 500일 때, 정확도는 k-means 대비 k-means++가 0.043 향상되었고, 1000일 경우는 3.308에서 3.31로 0.003 향상함을 확인했다. (iteration 50)

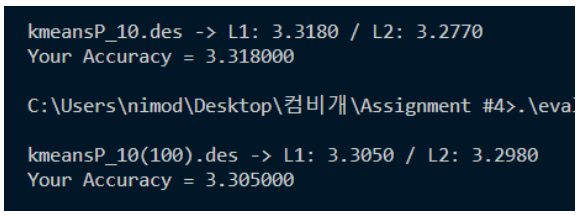


위:k-means / 아래:k-means++
seed=500

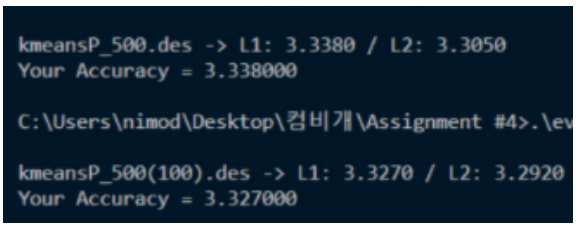


위:k-means / 아래:k-means++
seed=1000

iteration을 두 배 늘려 k-means++ 알고리즘을 적용해보았으나, 오히려 정확도가 감소한 것을 확인했다.



위: iter=50 / 아래: iter=100
seed=10



위: iter=50 / 아래: iter=100
seed=500
* iteration=50에서 가장 높은 정확도

iteration이 30, 40, 50, 60 일 때의 정확도는 다음과 같으며, iter가 50일 때 가장 높은 정확도를 기록한 것을 확인했다.

```
kmeansP_500(30).des -> L1: 3.3250 / L2: 3.2560
Your Accuracy = 3.325000

C:\Users\nimod\Desktop\컴비개\Assignment #4>.\ev

kmeansP_500(40).des -> L1: 3.2780 / L2: 3.2410
Your Accuracy = 3.278000
```

위: iter=30 / 아래: iter=40
seed=500

```
kmeansP_500.des -> L1: 3.3380 / L2: 3.3050
Your Accuracy = 3.338000

C:\Users\nimod\Desktop\컴비개\Assignment #4>.\ev

kmeansP_500(60).des -> L1: 3.2990 / L2: 3.2730
Your Accuracy = 3.299000
```

위: iter=50 / 아래: iter=60
seed=500