assn1 report

20160463 성해빈

Objective

주어진 n data point들을 k 개의 set으로 Mean Squared Error이 최소가 되도록 clustering 하는 것이 목표다. k는 디폴트로 5로 한다.

정확하게는 각 data point가 속하는 set의 mean과의 squared euclidian distance의 합이 최소가 되는 set(clustering)을 찾는 것이 목표다.

Method and Algorithm

Initialization Step (initialize_centroid)

k-means의 경우 n개 중에서 random하게 k개의 datapoint를 뽑고 그것들을 cluster의 mean(centroid)로 설정한다.

k-means++의 경우 처음에는 uniformly random하게 cluster mean(center)을 한 개 고르고, 고른 center와 각 data point의 distance의 제곱의 weight를 가진 probability distribution에서 data point를 한 개씩 뽑아 center를 하나씩 결정한다. 각 center를 뽑을 때마다 새로 뽑은 center에 대한 distance의 제곱으로 probability distribution weight를 업데이트한다.

Assignment Step (assign_points_to_cluster)

각 data point에 대해 가장 cluster mean(centroid)와의 least squared Euclidian distance가 작은 cluster를 고르고, 그 cluster를 data point에 assign 해준다. 이는 assigned_cluster에 `(index, cluster)` 값을 넣어주는 것으로 구현된다.

Update Step (update_centroid)

새로 assign한 data point들에 해당하는 cluster mean을 새로 계산해준다. 이걸 끝내면 또 (assignment step - update step)을 계속 수행해 준다.

Calculate tolerance (calculate_tolerance)

(assignment step - update step) 수행은 cluster set이 아예 안 바뀔 때까지 하는 것이 이상적이지만, 여기서는 tolerance 값을 계산해서 좀 더 유도리 있게 algorithm termination을 판단한다.

tolerance값은 현재 centroid와 이전 centroid의 distance를 이전 centroid의 length로 나눈 값을 합해서 계산해면 되는데, cluster mean의 변화율을 본다고 생각하면 될 것 같다.

이 tolerance값이 우리가 넣은 hyperparameter인 relative tolerance stop criteria보다 작은 경우에 알고리즘을 종료한다. criteria 디폴트는 0.001이다.

Discussion

k-means의 랜덤성 때문에 clustering 결과가 계속 바뀌어 내가 알맞게 구현했는지 알 수 없어 조금 불안하다. 하지만 centroid 값들이 예시와 같은 경우가 있어 아마 맞게 한 것 같다.

index가 꼭 data.txt의 순서와 일치한다는 말이 없다는 걸 깨닫고 일단은 index값도 assigned_cluster의 정보에다 넣게 바꿨다. data.txt에서 몇번째 줄인지와 상관없이 data.txt에 쓰여진 index값을 기준으로 data point들이 어느 cluster로 갔는지 알 수 있고, result.txt에도 data.txt에 쓰여진 index값을 기준으로 적힌다.

Time spent on assignment

About 3 hours