

Read Me

Hellogoogole - 12223544문규원, 12223557 여예진

실행방법

1. 실행방법-

1. '코드 파일 다운로드 후 구글 코랩에서 실행
2. 데이터셋 파일 구글 드라이브에 업로드
3. 전체 코드 실행 -> 파일 경로 확인

2. 세부 실행 방법

1. 구글 드라이브 마운트
2. Umap-learn 설치 (pip install umap-learn)
3. __main__에 arcne 데이터셋 파일 경로 지정, (전체 코드 마지막 부분)
4. 코드 실행

코드설명

1. 데이터 로드 및 전처리

DataLoader 클래스

init 메서드: 데이터 파일 경로를 초기화합니다.

load_data 메서드: 데이터를 로드하고 훈련 데이터와 검증 데이터를 합쳐서 반환합니다.

standardize_data 메서드: 데이터를 표준화합니다.

2. QKULDA 클래스

init 메서드: UMAP, lambda1, lambda2, reg_param, random_state 등의 초기 파라미터를 설정합니다.

fit_transform_umap 메서드: UMAP을 사용하여 데이터를 차원 축소합니다.

compute_rbf_kernel 메서드: RBF 커널을 계산합니다.

klda 메서드: Kernel LDA를 수행하여 데이터의 차원을 축소합니다.

calculate_Q1, calculate_Q2 메서드: Q1, Q2 값을 계산합니다.

calculate_Q 메서드: 최적화 목적 함수를 계산합니다.

optimize_sigma 메서드: 최적의 sigma 값을 찾습니다.

fit 메서드: LDA를 학습합니다.

predict 메서드: 학습된 LDA 모델을 사용하여 예측합니다.

cross_validate 메서드: 교차 검증을 수행합니다.

find_best_params 메서드: 최적의 lambda1, lambda2 값을 찾습니다.

fit_lda 메서드: 일반 LDA를 학습합니다.

predict_lda 메서드: 일반 LDA 모델을 사용하여 예측합니다.

3. 시각화 함수

plot_lda 함수

plot_lda 메서드: LDA 결과를 시각화합니다.

4. 실행 코드

main 함수: 전체 실행 코드를 포함합니다.

DataLoader 인스턴스 생성: 데이터 경로를 설정하여 인스턴스를 생성합니다.

데이터 로드 및 표준화: 데이터를 로드하고 표준화합니다.

UMAPKernelLDA 인스턴스 생성: UMAP과 Kernel LDA를 위한 인스턴스를 생성합니다.

UMAP으로 차원 축소: 데이터를 UMAP을 통해 차원 축소합니다.

교차 검증 수행: 차원 축소된 데이터를 사용하여 교차 검증을 수행합니다.

시각화: 최종 LDA 결과를 시각화합니다.

전체 과정 요약

데이터 로드: DataLoader 클래스를 사용하여 데이터를 로드하고 표준화합니다.

차원 축소: UMAPKernelLDA 클래스를 사용하여 UMAP으로 데이터를 차원 축소합니다.

Kernel LDA 학습 및 예측: UMAPKernelLDA 클래스를 사용하여 Kernel LDA를 학습하고 예측합니다.

교차 검증: 최적의 파라미터를 찾기 위해 교차 검증을 수행합니다.

결과 시각화: 최종적으로 LDA 결과를 시각화합니다.

QKULDA(Q-Kernel Optimized UMAP Linear Discriminant Analysis)는 UMAP을 사용하여 차원 축소를 수행하고, Kernel LDA를 사용하여 분류 성능을 극대화합니다. 특히 Q 최적화를 통해 성능을 향상시키는 것이 특징입니다.