LDA PROJECT

Team : KS

Member : 인공지능공학과 12194114 구현규

신소재공학과 12201068 손현제

Date : 2024. 06. 24

목차

1. 표지 .................................................... 1

2. 목차 .................................................... 2

3. 서론 ....................................................

3.1 LDA 소개

3.2 LDA 장/단점 분석

4. lda 개선.........................................

4.1 lda 개선

4.2 데이터셋

5. 결과 ........................................

6. 결론 ....................................................

7. 참고자료 ………….

**3. 서론**

**3.1 LDA 소개**

LDA란 지도학습의 한가지 방법으로, 두개 이상의 클래스가 존재할 때, 이를 선형적으로 분할하는 방법이다. 즉, 클래스의 차원을 줄이고 최대한 잘 구분하는 선형 변환을 찾아내는 것이 목표이다.

LDA는 다음과 과정을 거친다.

1. 클래스별 평균과 공분산 행렬 추정

각 클래스에 대해 평균 벡터와 공분산 행렬을 계산한다. 이 때 각 클래스의 데이터가 다변수 정규 분포를 따른다고 가정한다.

다변수 정규분포를 따른다고 가정하는 이유는 다음과 같다.

* 데이터 분포 가정의 단순화가 가능하다.

특정 변수에 대해서 최적화 하는 것이 목적인 데이터 분석 문제에서, 파라미터를 가장 simple하게 설정하는 것이 데이터 분석에 무조건적으로 유리하기 때문이다.

* 통계적 효율성을 높일 수 있다.

정규 분포의 가정을 통해, MLE를 이용한 평균 벡터와 공분산 행렬의 최적화 가 가능하다. 이렇게 되면 통계적 효율이 높은 분류기를 생성할 수 있게 한다.

1. 공통 공분산 행렬 추정

모든 클래스에 대해 공통 공분산 행렬을 계산한다. LDA는 각 클래스 간에 동일한 공분산 행렬을 갖는다고 가정한다. 공분산 행렬을 동일하게 가정하는 이유는, 모델의 단순화 및 계산 효율성 향상에 있다. 공통 공분산 행렬을 가정하면, 결정 경계를 선형으로 결정할 수 있다. 이는 LDA 계산에 있어서 간단하고 효율적으로 바뀐다는 것을 의미한다. 또한, 모든 클래스에 공통 공분산 행렬을 가정함으로써 공분산 행렬의 추정을 더 안정적으로 할 수 있다.

텍스트, 폰트, 친필, 화이트이(가) 표시된 사진

자동 생성된 설명LDA에서, 단순하게 두 집단만 가정하고 각각의 집단에 속할 확률을 생각 해봄으로써 이를 더 자세히 살펴볼 수 있다. 이중클래스에서 두 클래스를 완전하게 구분하기 위해 log 함수에 사후확률을 넣어서 사용할 수 있다. 공통 공분산 행렬을 가정하였기 때문에, 공분산에 같은 값을 넣고, 식을 전개할 수 있다. 텍스트, 폰트, 친필, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 폰트, 친필, 화이트이(가) 표시된 사진

자동 생성된 설명

폰트, 텍스트, 친필, 서예이(가) 표시된 사진

자동 생성된 설명최종적으로 얻어낸 식은 다음과 같고, 이를 일반화 하면 선형 결정 경계를 통한 판별 함수는 다음과 같이 나타낼 수 있다.

이 과정에서 수학적인 이득은 어떤 것이 있는지도 한 번 살펴보면, 공통 공분산 행렬을 가정하면, 공분산 행렬의 역행렬 를 한 번만 계산하면 된다. 이는 계산 비용을 크게 줄여준다. 만약 개별 공분산 행렬을 사용하는 경우, 각 클래스마다 공분산 행렬의 역행렬을 계산해야 하므로 계산 비용이 증가한다.

텍스트, 폰트, 화이트, 친필이(가) 표시된 사진

자동 생성된 설명다음으로, 클래스 간 분산을 최대화하고 클래스 내 분산을 최소화하는 방향으로 데이터를 변환하는 최적화 문제가 단순 해진다. (사진과 같은 형태)

여기서 는 클래스 간 산포 행렬, ​ 는 클래스 내 산포 행렬이다. 공통 공분산 행렬 Σ를 사용하면, ​ 단일 공분산 행렬로 표현되며, 이 최적화 문제를 일반화된 고유값 문제로 변환할 수 있다. 이는 해석과 계산이 더 용이한 형태로 변환된다.

1. 판별 함수 생성

폰트, 텍스트, 친필, 서예이(가) 표시된 사진

자동 생성된 설명위의 가정을 바탕으로 각 클래스에 대한 판별 함수 생성한다. 입력 데이터 포인트가 각 클래스에 속할 가능성을 평가한다.

1. 폰트, 텍스트, 화이트, 서예이(가) 표시된 사진

   자동 생성된 설명클래스 할당

새로운 데이터 포인트가 주어지면, 각 클래스의 판별 함수를 사용하여 가장 높은 값을 갖는 클래스로 데이터를 할당한다. argmax 함수를 통해서 가장 큰 값을 가지는 인덱스를 찾을 수 있다.

즉, 새로운 데이터 포인트 x가 주어졌을 때, 각 클래스 k에 대한 판별 함수를 계산하여 x가 클래스 k에 속할 가능성을 찾고, 그 확률이 가장 큰 클래스에 데이터를 할당한다.

**3.2 LDA 장/단점 분석**

**LDA 장점**

LDA는 클래스 간 분리를 최대화하는 선형 변환을 찾기 때문에 분류 문제에서 높은 성능을 발휘한다. 이 과정에서 데이터의 공통된 특성과 각 클래스의 평균을 고려하여 분류 경계를 설정한다. 이를 통해 데이터가 소속될 클래스의 결정 경계가 명확해지고, 분류의 정확도가 높아진다. 또한 LDA는 고차원 데이터를 저차원 공간으로 투영하면서 중요한 분산을 보존한다. 예를 들어, 3차원의 데이터를 2차원으로 축소하여 시각화 할 때, 데이터의 주요 구조를 유지할 수 있다. 이는 특히 고차원 데이터에서 유용하며, 차원 축소를 통해 계산 효율성을 크게 향상시킬 수 있다.

또한LDA는 다른 모델에 비해, 비교적 매개변수 수가 적고 구조가 간단한 모델이다. 이는 모델의 해석을 용이하게 하고, 과적합의 위험을 줄여준다. 간단한 선형 판별 함수로 인해 모델이 복잡하지 않아서, 데이터를 잘 설명하면서도 과도한 학습을 방지할 수 있다. 또한 LDA는 선형 판별 기법으로, 계산이 비교적 빠르고 효율적이다. 대규모 데이터셋에서도 빠르게 적용할 수 있으며, 실시간 분류 작업에도 적합하다. 이는 특히 실시간 데이터 분석이 필요한 응용 분야에서 유리하다.

다음으로, 데이터를 선형적으로 변환하여 클래스 간 분리를 최적화함으로써 데이터가 선형적으로 구분될 수 있는 경우에 매우 안정적인 성능을 제공한다. 이는 데이터의 선형적 특성을 잘 반영하여 일관된 결과를 도출하는 데 도움이 된다.

LDA는 여러 분야에서 활용될 수 있다. 예를 들어 얼굴 인식 분야에서는, 얼굴 이미지의 특징 벡터를 추출하여 사람을 구분하는 데 사용한다. 또한 문서 분류와 같은 분야에서, 텍스트 데이터를 분류하여 문서의 주제나 카테고리를 결정하는 데 사용한다. 이 외에도 금융 분야나 의료 분야에서도 유용하게 사용된다.

그리고, LDA는 다중 클래스 문제에도 적용할 수 있다. 이를 위해 각 클래스 간의 분산을 최대화하고, 클래스 내 분산을 최소화하는 방향 벡터를 찾는다. 다중 클래스 데이터를 효과적으로 구분할 수 있는 선형 판별 함수를 제공하여, 여러 클래스 간의 관계를 잘 반영할 수 있다.

마지막으로, LDA는 베이즈 이론에 기반하여 데이터의 분포를 모델링한다. 이는 분류 결과를 확률적으로 해석할 수 있게 한다. 예를 들어, 새로운 데이터 포인트가 특정 클래스에 속할 확률을 계산할 수 있다. 이는 결과의 신뢰도를 평가하고, 불확실성을 관리하는 데 도움이 된다.

**LDA 단점**

LDA는 각 클래스가 공통된 공분산 행렬을 가지는 정규 분포를 따른다는 가정을 한다. 이는 각 클래스의 데이터가 비슷한 형태의 분포를 가지고 있어야 한다는 의미이다. 만약 데이터가 이러한 가정을 따르지 않는다면, LDA의 성능이 저하될 수 있다. 또한 LDA는 클래스 간의 분리가 선형적으로 가능하다고 가정한다. 즉, 데이터가 선형적으로 구분되지 않는 경우, LDA는 효과적으로 작동하지 않을 수 있다. 비선형 경계가 필요한 데이터셋에서는 SVM(Support Vector Machine)이나 신경망(Neural Networks)과 같은 다른 기법이 더 적합할 수 있다.

다음으로LDA는 노이즈와 이상치에 민감하다. 노이즈나 이상치가 존재하면 공분산 행렬의 추정에 큰 영향을 미쳐서 분류 성능이 떨어질 수 있다. 이는 LDA가 데이터의 통계적 속성에 매우 의존하기 때문이다. 그리고 LDA는 클래스별로 충분한 데이터 포인트가 필요하다. 데이터가 적을 경우, 공분산 행렬의 추정이 부정확해질 수 있으며, 이는 모델의 성능에 부정적인 영향을 미친다. 특히, 고차원 데이터에서 차원이 데이터 포인트 수보다 많으면 차원의 저주(Curse of Dimensionality) 문제에 직면할 수 있다.

또, 변수 간의 다중공선성 문제를 겪을 수 있다는 것도 단점으로 볼 수 있는데, 다중공선성이란 독립 변수들 간의 높은 상관관계를 의미하며, 이는 공분산 행렬의 역행렬 계산을 불안정하게 만들 수 있다. 다중공선성이 있는 경우, LDA의 성능이 저하될 수 있다.

그리고 LDA는 선형 변환을 기반으로 하기 때문에 비선형 데이터 구조를 처리하기 어렵다. 커널 LDA와 같은 확장 방법이 있지만, 이는 복잡성을 증가시키고 계산 비용을 높인다. 비선형 분포를 가진 데이터에서는 비선형 분류 기법이 더 효과적일 수 있다.

LDA는 클래스 불균형이 있는 데이터셋에서 성능이 떨어질 수 있다. 이는 각 클래스가 공통된 공분산 행렬을 가지면서 정규 분포를 따른다는 가정과 관련이 있다. 클래스가 불균형하면, 소수 클래스의 정보가 충분히 반영되지 않아 분류 성능이 저하될 수 있다.

**4.lda 개선**

**4.1 LDA 개선**

기존 lda에서 이상치를 최대한 억제하는 방향성으로 lda 개선을 시도했다. 특히나 이미지 데이터 셋을 사용함으로써 한 데이터를 결정하는데 다양한 값이 들어가기 때문에 노이즈나 이상치가 관여하기 매우 쉬운 환경이기 때문이다.

따라서 크게 두가지 방법을 고려하였다.

1. 행렬의 정규화

행렬의 정규를 통해 가장 간단하면서도 이상치의 특이 값을 효과적으로 제어 할 수 있다 생각하여 가장 먼저 시도했다. 일반적으로 LDA에서는 클래스 간 분산과 클래스 내 분산의 비율을 최대화하는 방향으로 선형 판별 벡터를 찾는다. 이 때, 클래스 간 분산 행렬이 특히 작은 고유 값을 갖는 경우, 이상치나 잡음이 높은 데이터 포인트가 포함되어 있을 가능성이 크다.

따라서 이상치나 잡음이 높은 데이터의 경우 행렬의 고유 값이 작아질 수 있는데, 이를 해결하기 위해 정규화를 통해 고유 값의 크기를 증가시키고, 이상치의 영향을 줄이는 것이 목표다.

알고리즘보다는 직접적으로 코드를 확인해보자면

class\_sc\_mat += ((row - mv) / LA.norm(row - mv)).dot(((row - mv) / LA.norm(row - mv)).T)

S\_B += n \* ((mean\_vec - overall\_mean) / LA.norm(mean\_vec - overall\_mean)).dot(((mean\_vec - overall\_mean) / LA.norm(mean\_vec - overall\_mean)).T)

다음과 같이 해당 행렬의 la.norm을 계산하여 해당 행렬의 값을 나누어 주는 것으로 행렬을 정규화 하였다. 기대 효과로는 이상치를 효과적으로 제어하여 분석 성능향상에 기여를 할 것으로 기대된다.

2. 가중 중앙값 사용

이상치가 LDA의 성능을 저하시키는 주요 이유는, 이상치가 클래스 내 샘플들의 평균에 큰 영향을 미치기 때문이다. 이를 해결하기 위해, 우리는 클래스 내 샘플들의 평균 가중 중앙값 로 대체하고자 한다. 는 다음과 같이 계산한다. i번째 클래스의 샘플 집합 =[,]가 주었을 때 각 변수에 가중치를 주어 이상치에 대처하고 그 값으로 평균을 대체하는 것이다.

=

먼저 가중치는 위와 같이 계산하는데 샘플 집합의 median 값을 이용해 벗어나 있는 만큼 중치를 부여하는 것이다. 이때 0으로 떨어지는 것을 방지하기 위해 아주 작은 값을 적용하여 대응했다.

=

다음은 결과적으로 사용하는 계산하는 방법으로 각 데이터에 가중치를 부여하고 이 모든 값을 총 가중치로 나누어 이상치 값이다.

위에서 볼 수 있듯이, 샘플이 해당 클래스에서 얼마나 벗어나 있는지를 계산하여 가중치 값을 계산하는데, 이는 결국 거리에 반비례하는 값으로 떨어져 이상치에는 작은 값이 곱해지게 된다. 결과적으로 계산 과정에서 이상치의 영향을 줄일 수 있을 것으로 기대된다.

3. lda 투영행렬 결정

행렬 W는 의 특성 벡터를 구하는 것과 동일 한데, 샘플 분포가 다양할 때 샘플 분포가 클래스 내 집중된 특성을 가지면, 클래스 간 거리를 증가시키기만 하면 최적의 효과를 얻을 수 있습니다. 이 경우 행렬의 투영 행렬에 대한 영향을 배제하고, 클래스 간 분산 행렬 의 특성 벡터만을 구하여 투영 행렬 W로 사용하면, 행렬의 역행렬 계산을 피할 수 있어 LDA의 연산 복잡도를 크게 줄일 수 있다. 동일하게 샘플 분포가 이미 클래스 간에 분산된 경우, 행렬만을 투영 행렬로 사용할 수 있다. 이러한 처리는 클래스 내와 클래스 간 분산 거리의 크기를 알아야 한다.

또한 클래스 내 분산 행렬의 특성 값 크기는 본질적으로 클래스 내 분산 거리의 크기를 반영한다. 특성 값이 클 때, 클래스 내 샘플은 더 분산되고, 해당 거리는 더 크다. 동일하게, 클래스 간 분산 거리의 크기도 클래스 간 분산 행렬의 특성 값으로 나타낼 수 있다. 이를 이용하여 알맞은 특성 벡터를 구할 수 있다.

다음과 같은 규칙을 정하여 행렬을 결정한다

1.Trace()>10\*Trace()이면 클래스 내 분산 거리가 너무 분산되었음을 나타내므로, 클래스 내 분산 행렬 의 특성 벡터를 구하여 최적의 LDA 투영 행렬로 사용합니다.  
2.Trace()<0.1\*Trace() 클래스 내 분산 거리가 너무 분산되었음을 나타내므로, 클래스 내 분산 행렬 의 특성 벡터를 구하여 최적의 LDA 투영 행렬로 사용합니다.

3. 기타의 경우 원래의 특성벡터를 사용한다.

다음과 같은 규칙을 이용하여 최상의 경우 계산 량과 적절한 특성 벡터 값을 결정할 수 있을 것으로 기대된다.

**4.2 데이터셋1**

처음으로 선정한 데이터셋은 Balance Scale 특정 무게와 거리로 구성된 데이터셋으로, 주어진 무게와 거리에서 어느 쪽이 더 무거운지 예측할 수 있다. 총 3개의 클래스가 있고, 4개의 속성과 625개의 데이터가 존재한다.

-사용의 어려움

이 데이터 셋의 경우 클래스의 숫자도 많지 않고 데이터의 숫자도 많지 않으며, 각 데이터의 라벨도 많지 않아 이미 구현전에도 87.77 퍼센트의 정확도를 보이고 있다.

따라서 개선의 효과가 미미 할 것으로 예상된다.

**4.2 데이터셋 2**

LDA의 개선 전후의 정확도 차이가 없었던 원인을 예측해 보았을 때 Balance scale 데이터셋은, 규모가 너무 작았고 데이터셋 분류가 너무 잘되어 있어 기존 LDA로도 높은 성능을 낼 수 있었다고 생각했다. 또한 한데이터를 결정하는 라벨의 숫자가 많아 변수가 많은 데이터 셋이 필요했다

이런 피드백을 바탕으로, 새로운 데이터셋 설정을 통해 새로운 프로젝트를 진행하였다.,

다음으로 선정한 데이터셋은 Face Expression Recognition Dataset이다, 7066개의 사람 표정을 이미지로 가지고 있으며, angry, disgust, fear, happy, neutral, sad, surprise 총 7개의 클래스로 이루어져 있다. (7개의 클래스를 쓰는 경우 시간도 많이 걸리고 정확도가 너무 낮아 4개만 사용했습니다)

**5.결과**

**1. 데이터셋 1**

방법 1,3의 경우 87.77으로 성능향상이 보이지 않았다.

방법 2를 추가한 1,2,3의 경우 88.83으로 성능이 향상되었다.

**2. 데이터셋 2**

1. 평범한 lda를 이용하여 계산하면

LDA Classification Accuracy on validation set: 47.09%의 결과 값이 나온다.

2. 1+3 lda

LDA Classification Accuracy on validation set: 47.73%의 결과 값이 나온다

3. 1+2+3

**6.결론**

위 결과를 토대로 보자면 정규화와 3의 방법으로 이상치를 제어할 경우 성능을 향상시킬 수 있음을 확인할 수 있었다. 정규화의 경우 데이터 셋이 복잡할 경우 효과적이었으면 3의 경우 두 데이터 모두 효과적임을 알 수 있었다. 이상치가 생각보다 결과에 큰 영향을 미칠 수 있음을 알 수 있었다.

2의 경우 다른 특성 행렬을 사용하는 특이적인 경우가 잘 발생하지 않아 효과를 확인한기 힘들었다. 또한 계산의 복잡성 문제도 같은 이유로 성능 향상을 느끼기 힘들었다.

마지막으로 각 데이터 간의 공간적 특성이나 거리의 특성 값을 고려하는 방식으로 데이터의 정보를 좀더 연관성 있게 고려하는 방식도 고민했는데, 구현하지 못해 아쉬운 느낌이 있다.

**7.참고자료**

Guo, L., & Jin, L. (2010). “An improved Linear Discriminant Analysis method based on regularization.” In Proceedings of the International Conference on Pattern Recognition (pp. 2067-2070). IEEE.

Ioffe, S. (2006). “Probabilistic Linear Discriminant Analysis.” In A. Leonardis, H. Bischof, & A. Pinz (Eds.), ECCV 2006, Part IV, LNCS 3954 (pp. 531-542). Springer-Verlag Berlin Heidelberg.

코드 및 알고리즘 구현규

보고서 및 발표 손현제