핸즈온 머신러닝

PART5. 비선형 SVM 분류

1) 다항식 특성 추가: SVM을 사용할 땐 커널 트릭 적용.

- 커널 트릭: 실제로는 특성을 추가하지 않으면서 다항식 특성을 많이 추가한 것과 같은 결과를 얻을 수 있음.

2) 유사도 함수 특성 추가: 각 샘플이 특정 랜드마크와 얼마나 닮았는지 측정.
방사 기저 함수를 유사도 함수로 정의

- 방사 기저 함수: Radial Basis Function (RBF) kernel은 또한 Gaussian kernel이라고도 불리며,
이것은 커널인데, RBF의 형태를 취하는 것으로 더욱 구체적으로 가우시안 형태를 취하는 커널이 되겠다.
RBF 커널은 아래와 같이 정의 된다.

$$K_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left[-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right]$$

- 감마는 파라미터로 커널의 흩어짐을 세팅한다. 즉, 사실 가우시안 분포에서 보면, 분산에 해당하는 부분이므로, 흩어짐을 세팅한다고 할 때, 이해가 쉽게 갈 수 있다. 감마는 0보다 커야하며 값이 작을수록 폭이 넓은 종 모양이 된다. 즉 샘플이 넓은 범위에 걸쳐 영향을 주므로 결정경계가 더 부드러워진다. 반대로 감마가 커지면 종모양 그래프가 좁아져서 각 샘플의 영향 범위가 작아진다.
=> 모델이 과대적합일 경우 감마를 감소, 과소적합일 경우 감마를 증가
- 파이는 함수로써, 벡터 x를 새로운 벡터 공간으로 사영시키는 기능을 한다. 이 커널 함수는 두 개의 사영된 벡터들 사이에서 내적을 계산한다.

핸즈온 머신러닝

PART5. 비선형 SVM 분류

- 군집화, 이상치 분석, 최근접 이웃 분류와 같은 데이터 마이닝 응용분야에서 상호간 비교를 통해 어느 정도 같거나 다른지를 평가하는 방법이 필요함

○ 군집

- 데이터 객체의 집합

- 한 개 군집 내의 객체는 서로 유사함

- 다른 군집의 객체와는 서로 다른 점이 있음

- 이상치 분석은 군집 기반 기술을 사용하여 나머지 값과 상이한 객체를 잠정적 이상치로 구분

- 객체유사도에 대한 지식은 최근접 이웃 분류 스키마에서 사용됨

○ 유사도와 비유사도

- 유사도와 비유사도는 상호 관련이 있음

- 2개 객체 i와 j가 다를 경우 이에 대한 유사도 측정값은 0

- 유사도 값이 커질수록, 객체 사이의 유사한 정도는 큼(일반적으로 값 1은 완전일치를 말함)

- 비유사도는 반대의 성질

핸즈온 머신러닝

PART5. 비선형 SVM 분류

2.4.1 데이터 행렬 대 차이 행렬

- 이전 글에서 다뤘던 1차원 객체와 달리 다차원 속성으로 표현된 객체를 다룸
- n개의 객체(사람, 상품, 과정)는 p개의 속성(연령, 신장, 체중, 성별과 같은 측정값이나 특성)에 의해 기술한다고 가정
- 객체는 $x_1=(x_{11},x_{12},\ldots,x_{1p}),x_2=(x_{21},x_{22},\ldots,x_{2p})$ 등으로 나타냄
- x_{ij}는 객체 x_i의 j번째 속성 값
- 관계형 DB에서 객체는 튜플이며, 데이터 샘플이나 속성벡터라고 함
- 주 메모리 기반 클러스터링과 최근접 알고리즘은 다음 2개의 데이터 구조 중 한 개에 근거하여 실행함

(1) 데이터 행렬(객체와 속성 구조)
- n개의 데이터 객체를 관계형 표나 n행(객체)과 p열(속성)의 행렬 형태로 저장
- 각 행은 객체 1개에 해당
- 표기법의 일환으로 p개의 속성을 인덱스하기 위해 f개를 사용할 수 있음
- 데이터 행렬을 2개 모드 행렬이라고도 함

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

PART5. 비선형 SVM 분류

(2) 차이 행렬(객체와 객체 구조)
– n개 객체의 모든 쌍에 대해 근접도를 저장
– 종종 n대 n표로 표시

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ \dots & \dots & \dots & \\ d(n,1) & d(n,2) & \dots\ \dots & 0 \end{bmatrix}$$

– d(i, j)는 객체 i와 j간의 비유사도를 측정한 값
– 객체 i와 j가 유사한 경우 0에 가까운 음이 아닌 값을 가짐, 큰 값을 가질수록 차이가 큼
– d(i, j) = d(j, i) : 대칭행렬
– sim(i, j) = 1 – d(i, j) : sim(i, j)는 유사도
– 차이 행렬을 1개 모드 행렬이라고도 함

핸즈온 머신러닝

PART5. 비선형 SVM 분류

2.4.2 명목 속성의 유사도 측정

- 명목 속성은 2개 이상의 값을 가짐

ex) map_color는 명목 속성으로 빨강, 노랑, 녹색, 분홍, 파랑색과 같은 5개의 상태를 가짐

- 명목 속성의 상태 개수를 M개라고 할 때, 이 상태값은 1, 2, ..., M과 같이 알파벳 문자, 기호, 정수로 표시할 수 있음
- 정수로 표시할 경우 정수는 데이터 핸들링을 위해 사용하며 특별한 의미는 없음

○ 명목 속성을 갖는 객체 간 차이 계산

- 두 객체 i와 j 간 차이는 비일치 비율에 근거하여 계산할 수 있음

$$d(i, j) = \frac{p - m}{p}$$

- m은 일치하는 속성의 수(i와 j가 동일한 상태에 있는 속성의 개수)
- p는 객체를 표현하는 전체 속성의 수
- 가중치는 m개의 효과를 증가시키기 위해 사용함
- 유사도는 다음과 같음

$$sim(i, j) = 1 - d(i, j) = \frac{m}{p}$$

※ 명목 속성 객체 간 유사도는 인코딩을 하여 계산함

- M개의 상태에 대해 각각 이진값을 생성함으로써 비대칭 이진 속성을 인코딩함
- 특정 상태값을 갖는 객체에 대해 이진 속성값은 1이고 다른 이진 속성값은 0이 됨

ex) map_color를 인코딩하기 위해 이진 속성값은 5개의 색상에 대해 생성 가능

노란색 속성은 1로 설정, 나머지는 0으로 설정과 같이 인코딩

$$K_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left[-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right]$$

여기서 보면, 제곱에 해당하는 부분은 아래와 같이 분해될 수 있다.

$$= \exp\left[-\frac{1}{2}\langle \mathbf{x} - \mathbf{x}', \mathbf{x} - \mathbf{x}' \rangle\right]$$

즉, x - x' 이라는 두 개의 벡터를 내적한 것에 해당한다. 여기서 다시 분해해서, 아래와 같이 내적 결과들의 뺄셈으로 나타낼 수 있다.

$$= \exp\left[-\frac{1}{2}(\langle \mathbf{x}, \mathbf{x} - \mathbf{x}' \rangle - \langle \mathbf{x}', \mathbf{x} - \mathbf{x}' \rangle)\right]$$

여기서 다시 한번 분해시켜 보자. 각각의 원소들로 쪼개어 내적으로 표현하면 아래와 같다.

$$= \exp\left[-\frac{1}{2}(\langle \mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{x}, \mathbf{x}' \rangle - \langle \mathbf{x}', \mathbf{x} \rangle + \langle \mathbf{x}', \mathbf{x}' \rangle)\right]$$

$$= \exp\left[-\frac{1}{2}\|\mathbf{x}\|^2 - \frac{1}{2}\|\mathbf{x}'\|^2\right]\exp\left[-\frac{1}{2} - 2\langle \mathbf{x}, \mathbf{x}'\rangle\right]$$

위의 수식에서 특정한 x와 x'에 대한 norm의 값은 상수 값으로 아래와 같이 세팅 한다면,

$$C := \exp\left[-\frac{1}{2}\|\mathbf{x}\|^2 - \frac{1}{2}\|\mathbf{x}'\|^2\right] \text{ is a constant}$$

RBF kernel은 단순화 시키면 아래와 같다.

$$= Ce^{\langle \mathbf{x},\mathbf{x}'\rangle}$$

그런데 여기서 테일러 시리즈 expansion을 사용할 수 있다. exponential 함수의 형태를 취하기 때문이다.

$$= C\sum_{n=0}^{\infty}\frac{\langle \mathbf{x}, \mathbf{x}'\rangle^n}{n!}$$

핸즈온 머신러닝

PART5. 비선형 SVM 분류

1) 다항식 특성 추가: SVM을 사용할 땐 커널 트릭 적용.

- 커널 트릭: 실제로는 특성을 추가하지 않으면서 다항식 특성을 많이 추가한 것과 같은 결과를 얻을 수 있음.

2) 유사도 함수 특성 추가: 각 샘플이 특정 랜드마크와 얼마나 닮았는지 측정.
방사 기저 함수를 유사도 함수로 정의

- 방사 기저 함수: Radial Basis Function (RBF) kernel은 또한 Gaussian kernel이라고도 불리며,
이것은 커널인데, RBF의 형태를 취하는 것으로 더욱 구체적으로 가우시안 형태를 취하는 커널이 되겠다.
RBF 커널은 아래와 같이 정의 된다.

⇒ 데이터 셋에 있는 모든 샘플 위치에 랜드마크 설정.
⇒ 장점: 차원이 매우 커지고, 따라서 변환된 훈련 세트가 선형적으로 구분될 가능성이 높다.
⇒ 단점: 훈련 세트에 있는 n개의 특성을 가진 m개의 샘플이 m개의 특성을 가진 m개의 샘플로 변환
(동일한 크기의 아주 많은 특성이 만들어져 연산 비용이 만들이 든다)
=> 보완: SVM에 가우시안 RBF 커널 사용한 SVC 모델

핸즈온 머신러닝

PART5. 비선형 SVM 분류

- 여러가지 커널 중에 선형 커널을 가장 먼저 시도해봐야 한다. (Linear SVC > SVC(kernel="Linear"))
- 훈련 세트가 너무 크지 않다면 가우시안 RBF 커널도 시도해보면 좋다.
- 시간과 컴퓨팅 성능이 충분하다면 교차 검증과 그리드 탐색을 사용해 다른 커널도 좀 더 시도.

[계산 복잡도]
1. Linear SVC 파이썬 클래스는 선형 SVM을 위한 최적화된 알고리즘을 구현하 liblinear 라이브러리를 기반. 커널 트릭은 지원하지 않지만 훈련 샘플과 특성 수에 거의 선형적으로 늘어난다. 훈련 시간 복잡도는 대략 O(m x n) 정도.
- 정밀도를 높이면 알고리즘의 수행 시간이 길어진다. 이는 허용오차 하이퍼파라미터 엡실론으로 조절 (사이킷런에서는 매개변수 tol). 대부분의 분류 문제는 허용오차를 기본값으로 두면 잘 작동.

⇒ SVC의 tol 매개 변수 기본값은 0.001이고, LinealSVC의 tol 매개변수 기본값은 0.0001이다.

2. SVC는 커널 트릭 알고리즘을 구현한 libsvm 라이브러리를 기반. 훈련의 시간 복잡도는 보통 O(m^2 x n) 과 O(m^3 x n) 사이이다. 훈련 샘플 수가 커지면 엄청나게 느려진다.

⇒ 복잡하지만 작거나 중간 규모의 훈련 세트에 이 알고리즘이 잘 맞음

⇒ 특성의 개수(희소 특성: 각 샘플에 0이 아닌 특성이 몇 개 없는 경우)인 경우에는 잘 확장. 이 때는 알고리즘의 성능이 샘플이 가진 0이 아닌 특성의 평균 수에 거의 비례