1. **논문 요약**

이 논문은 대규모 데이터를 SVM으로 분류할 때 고려해야 할 점에 대해 논하고 있다. 즉, SVM 최적화 방법을 설명한다. 여기서는 대규모 SVM에서 더 메모리와 시간복잡도 측면에서 더 실용적인 을 소개한다.

최적화 문제는 training data 수인 에 의존한다. 행렬Q의 크기가 이기 때문에 이 10000만 되도 메모리상, 시간복잡도상 매우 비효율적이다. 이런 대규모 데이터에 SVM를 적용하기 위해서는 이를 최적화하는 작업이 필요하다.

접근법 중 하나로 데이터를 분해하는 방법이 있다. 는 Osuna et al의 분해 아이디어를 사용한다. 하지만 이 방법은 training 시간이 오래 걸린다는 단점을 극복해야 한다. 이 논문은 다음과 같은 대안으로 다음과 같은 방법을 제시한다.

* 효과적이고 효율적인 working set을 선택하는 방법
* 최적화문제의 성공적인 축소
* 캐싱 및 그라디언트와 종료 기준과 같은 전산적 발전

첫번째 논문: 목적-> 거대한 범위의 SVM을 어떻게 만들것인가?

SVM

problem

1.경계선의 한계와 선형 등가제약 조건등 과 같은 2차 최적화 면에서 결함이 있다.

2.실무적인 상황에서 일반적인 2차적 프로그램을 최적화시킬떄에도 메모리와 시간문제로

다루기 힘들어진다.

추가조사

장점 :

그것은 분리의 명확한 여백으로 정말 잘 작동합니다.

높은 차원 공간에 효과적입니다.

차원 수가 표본 수보다 큰 경우에 효과적입니다.

결정 기능 (지원 벡터라고 함)에서 학습 점의 하위 집합을 사용하므로 메모리 효율도 높습니다.

단점 :

필요한 교육 시간이 더 길기 때문에 대용량 데이터 세트를 사용할 때 성능이 좋지 않습니다.

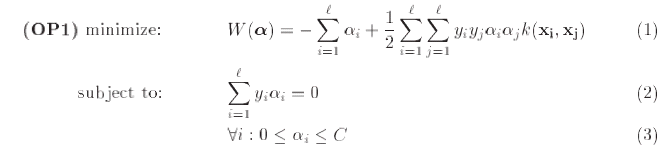
또한 데이터 세트가 더 많은 잡음을 가지고있을 때 (즉, 대상 클래스가 중첩되는 경우) 잘 수행되지 않습니다.

SVM은 확률 산정을 직접 제공하지 않으며, 이는 비싼 5 배 교차 검증을 사용하여 계산됩니다. Python scikit-learn library의 SVC 관련 메소드입니다.

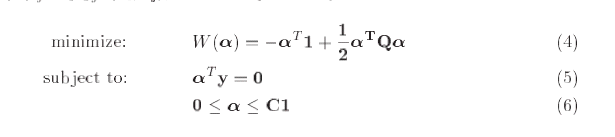
solution

SVMlight: 거대한 작업량을 처리할때 사용하는 SVM용

"" V 2.0 : large-scale SVM training machine->넓은 범위의 데이터를 사용할 수 있는 SVM을 구축할 수 있다.



식1



식2

식1-> 식2:

식1은 일반적인 SVM에서 발생하게 되는 2차원적인 최적화과정에서 구하는 방식이다,

이것을 최대한 좋은 방향으로 솔루션을 찾아본 결과{(1)을 최소화되는 지점이 최고의 지점, (2),(3)은 반드시 유지되어야 하는 수치이다). 문제는 최적의 솔루션으로 가기도 전에 l(size of data)가 너무 크면 Q(a들자표집단)을 메모리에 저장하고 있을 수 없다. 다른 방법이 있겠으나, 들여야 되는 수고가 너무 크다.

그래서 이런 문제를 해결하기 위해서 처리해야할 큰 문제를 조금씩 잘라서 해결하는 일이다

-> SVM light는 이런 일에 사용된다,

-> OP1은 조금만 크기 여러 개로 나누어지어 메모리를 요구로 하는 여러 알고리즘으로 이루어진다.(실질적으로 작업을 하는 부분"working part"를 중요시 여기면 된다)

-> 다만 training 시간이 조금 많이 걸리게 된다.

정리: 그냥 SVM을 사용하게 되면 데이터의 용량이 커졌을 때 메모리가 감당할 수 없을 것을 고려하여 문제를 쪼개어 생각하는 알고리즘을 짜도록 한다.

두 번째 논문

(서론)

번역,회전 및 스케일링을 포함한 감각적인 프로파일링을 통하여 데이터를 다룸.

\*프로파일링: 데이터 품질 측정 대상 데이터베이스의 데이터를 읽어 컬럼, 테이블의 데이터 현황정보를 통계적으로 분석하

전반적인 데이터 매트릭스의 합의점(?)를 구하기 위해 GPA와 STATIS의 방법을 비교하면서 사용한다.

그 두가지 방법 사이에서 임의적인 오류와 실제의 근본 합의의 의의를 통해서 중요한 성능 차이를 찾아내려고 했다.

그러나 각 방법의 평균치수는 일부 이론에 의해서만 뒷받침되고, 문제사항이 너무 빈번하여 양쪽(static, GPA)의 방법을 모두 사용하는 접근식을 통해서 시뮬레이션에서 더 좋은 결과를 얻었다.

(필요성)

미지의 영역의 감각적인 데이터의 프로파일링에서의 세 가지 문제

1.사용자마다 다른 범위를 이용한다는 점

2. 사용자들의 특징에 대한 혼란

3.zero 에대한 다양한 의견

-> statistical method가 이런문제들을 해결하기 위해 발명

(여러 감각적인 데이터들의 매트릭스들의 합의점(\*)을 얻기 위해)

\*합의점: 데이터 집단의 진짜 데이터의 구조와 그 데이터들간의 상호관계를 알려주어야 한다.

(결론)

두 가지 방법들은 프로파일링 데이터에서 고정된 방식 뿐만 아니라 다양한 선택으로 사용될 수

있다.

그룹의 평균이 합의점과 유사하게 추출될 수 잇는 방법을 얻기 위해서는 유사성에 대해서 정의해여할 필요가 있었다. 이론적인 평균의 배분은 어렵게 보였기, monte-carlo-simulation의 방식을 통해서 평균의 배분을 정의했다.

(\*monte-carlo-simulation?) 자료조사

-확률이 존재하는 의사결정상황(많은 결과를 정확하게 예측할 수 없는, 변수의 관계가 불분명한 상황)에서 수치적으로 일련의 난수를 반복적으로 발생하면서 시뮬레이션을 돌려서 답을 얻어내는 방식-> 단위 변수의 패턴을 알아내고, 그 단위의 변수의 조합의 패턴을 추정->> 각각의 변수의 확률 분포를 통합해서 목표로 하는 값의 확률분포를 알아내는 것

출처: <http://egloos.zum.com/yjhyjh/v/33072>

정리: static과 GPA의 방법을 융합시킨 방법으로 데이터들의 협의점과 경계를 찾는 simulation이 가장 좋은 방법이다.