워크북

교과목명 : 머신 러닝

차시명: 12차시

◈ 담당교수: **장 필 훈**

⊙ 세부목차

- 순차데이터
  - LDS
  - 칼만필터
- 모델조합
  - bagging
  - adaboost
  - gradient boost

학습에 앞서

■ 학습개요

순차데이터의 마지막 시간으로 선형동적시스템을 배운다. 대표적으로 칼만필터를 보고, 어떤식으로 학습하는지 이전에 배운것들을 바탕으로 이해한다.

개별적 모델들의 합인 모델 조합에 관해 배운다. 기본적인 모델조합의 아이디어부터, 배깅, 부스팅에 관해 자세히 배운다. 배깅은 기본적으로 단순히 여러모델의 voting으로 이해할 수 있는데, 이때 오류가 어떻게 변하는지 식으로 확인한다. 다음으로는 부스팅에 관해 배운다. 부스팅은 여러종류가 있으나 가장 널리 알려진 에이다부스트를 자세히 배우게 된다. 학습알고리즘부터, 수식에 관해 자세히 분석하고 결과를 시각화해서 어떤식으로 분류가 이루어지는지 확인한다. 다음으로 그래디언트 부스팅에 관해 배운다. 그래디언트 부스팅은 실제로 가장 많이 쓰이는 부스팅이므로 간단한 예제를 통해 바로 적용이 가능하도록 배운다.

■ 학습목표

1	선형동적 시스템의 개념을 대략 이해한다
2	선형동적 시스템의 대표적인 예인 칼만필터를 정성적으로 이해한다.
3	adaboost의 학습과정을, 해석과 함께 수식으로 자세히 이해한다.
4	gradient boosting을 배우고 예제를 통해 숙지한다.

■ 주요용어

용어	해설
칼만 필터	잡음이 포함되어 있는 측정치를 바탕으로 선형 역학계의 상태를 추정하는 재귀 필터. (@wiki)
부트스트랩	현재 있는 표본에서 추가적으로 표본을 복원 추출하고 각 표본에 대한 통계량을 다시 계산하는 것 (@wiki)
adaboost	adaptive boosting을 줄인 말. weak learner여러개를 조합해서 강력한 분류기를 얻는다는 아이디어에 기본하며, 이 과정에서 데이터포인트들의 weight를 adaptive하게 조정하는데, 'ada'는 이 'adaptive'에서 나옴.
gradient boost	부스팅의 한 종류. adaboost처럼 연속적으로 학습하되, 잔차(residual)에 fit한다. 이 과정에서 gradient를 이용하므로 gradient boosting이라고 함.

학습하기

선형동적시스템은 굉장히 직관적입니다. 우리가 맨 처음에 배웠던 선형회귀와 근본적으로 동일합니다.

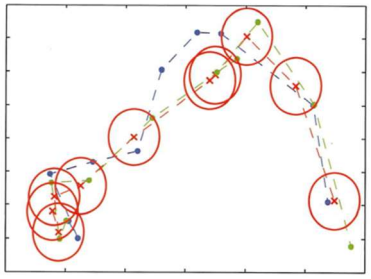
순차 데이터에서 선형예측으로 생각하기 가장 쉬운것은 최근값들의 선형함수일 것입니다. 더 단순하게는 가장 최근값의 반복도 생각해볼 수 있습니다. 모두 선형동적 시스템 맞습니다.

만일 분포라면 어떻까요. 데이터가 계속 들어오면서 분포가 일정하게 변한다고 하면 어떤식으로 기술할 수 있을까요. 일단 곱셈에 대해 닫혀있는 분산이 필요할겁니다. 연쇄의 길이가 길어질수록 복잡도가 증가하지 않는다는 좋은 특성을 가졌습니다. 그리고 곱셈에 대해 닫혀있는 분포는 지수족이 있습니다. 이 추론문제를 푼 것이 칼만필터입니다.

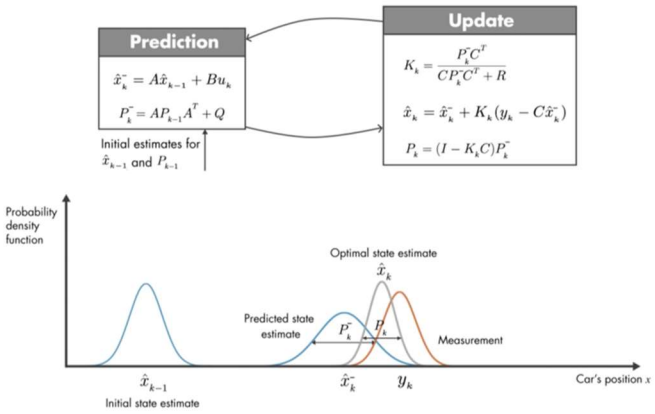
칼만필터는 기본적으로 선형시스템이므로 이전상태를 바탕으로 다음상태를 추측해 내는 것이 목적입

니다. hmm과 같이 잠재변수와 분포를 가정하고, 시간적 변화에 따른 선형변환과 노이즈를 반영합니다. 물론 잠재변수와 노이즈는 모두 가우시안을 가정합니다.(가우시안이 지수족 분포)

우리가 실제로 계산해보지는 않겠지만, 일단 직관적으로는 다음 그림(Bishop, Fig 13.22)을 상상하면 좋습니다.



2차원에서 물체의 실제 위치를 예측해보는 것입니다. 물론 우리의 관측은 실제값+노이즈 입니다. 그림에서 파란점이 실제위치, 녹색점이 관측값(실제값+노이즈), 붉은점이 칼만필터의 예측값(추론사후분포의 평균)입니다.



왼쪽그림은 제가 칼만필터 공부하면서 봤던 그림중에 가장 짧고 요약이 잘된 그림이어서 참고하시라고 넣었습니다. (YouTube에서 ‘, Matlab, Understanding Kalman Filters, Part 4’로 검색하시면 됩니다. 매틀랩공식채널입니다.) 모델의 상세를 보신다거나 구현이 필요할 때 참고하면 좋습니다.

<모델조합>

모델조합은 말 그대로 여러가지 모델을 조합해서 쓰는 것입니다. 주의할 점이 있다면, 여러가지 모델의 평균값 혹은 그 함숫값으로 데이터포인트가 주어지는 것이 아니라, 여러가지 모델중의 하나에서 데이터포인트가 생성되었다는 뜻입니다. 즉, 데이터포인트중에 특정 포인트를 추출해서 보면, 그것이 여러 모델의 평균값이 아니라, 조합된 여러 모델중 하나로부터 추출되었다는 뜻입니다.

<bagging>

배깅은 bootstrap aggregating의 준말입니다. 큰 데이터집합에서, 일정수의 샘플집합을 얻는 과정을 반복하고 각 샘플집합에 대해 학습하는 방법입니다. 최종예측은 각 모델이 주는 값을 voting해서 정하는 것이 가장 일반적입니다. 트리와 같이 불안정성이 큰 분류기에 효과가 큽니다. 하나의 트리가 불안정한 값을 주더라도 트리가 100만개쯤 되면 안정적인 결과가 나오겠지요. 이 경우 완전히 이상적인 경우 bias와 variance를 모두 줄이는데 왜 그렇게 되는지를 보이겠습니다. 그리고 실제로 왜 그렇게 잘 안되는지도 알아보겠습니다.

예측하고자 하는 함수를 h(x)라 하면 출력은 y(x)=h(x)+e(x)이고 , 제곱합 오류는

$E[(y(x)-h(x))^2] = E[e^2(x)]$가 되고, 개별적으로 M개의 모델을 시험했을 때 평균오류는

$$E = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_x[\epsilon_m(\mathbf{x})^2]$$ 왼쪽과 같습니다. 독립적 모델의 1/M에 불과하다는 뜻입니다. 각각의 모델을 썼을 때 평균보다 1/M에 불과하고 데이터는 추가적으로 필요하지 않다는 결과는 굉장히 놀랍습니다. 놀라운 만큼 비현실적이지요. 보통 오류는 높은 상관관계를 가집니다. 그래서 이렇게 극적인 결과는 잘 일어나지 않습니다. 다만, 줄어들기는 줄어듭니다. 아주 약간의 성능향상을 가져오지요. 그래서 계산용량이 허용하는 한 많은 모델을 조합해서 쓰는것이 성능에는 더 좋습니다.

<adaboost>

boosting중 가장 널리 쓰이는것중 하나인 adaboost입니다. 부스팅의 기본 아이디어는 'weak learner(base classifier)여러개를 모아서 더 좋은 분류기를 만들수 있다'는 것입니다. 'weak learner'란 성능이 좋지 않은(맞출 확률이 반이 간신히 넘는) 모델을 말합니다.

여러개의 base classifier는 sequential하게 배열됩니다. 맨 앞의 분류기가 학습하고 나면, 그 결과에 따라 데이터포인트의 weight가 결정되고 그 중요도에 따라 두번째 분류기가 학습합니다. 이렇게 계속 반복하다보면 strong learner를 얻을 수 있습니다. 마지막 분류기의 출력값이 전체 분류기의 출력값이 되는것이 아니라, 모든 분류기의 출력값들의 weighted sum이 최종 출력값이 됩니다. 구체적인 과정은 다음과 같습니다.

(1) 가중치 초기화 (w_n = 1/N)

(2) 오류함수 최소화. 오류함수 $J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)$. I: indicator function

(3) 다음계산반복

$$\alpha_m = \ln\frac{1-\epsilon_m}{\epsilon_m}, \qquad \epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\}$$

(4) 최종결과:

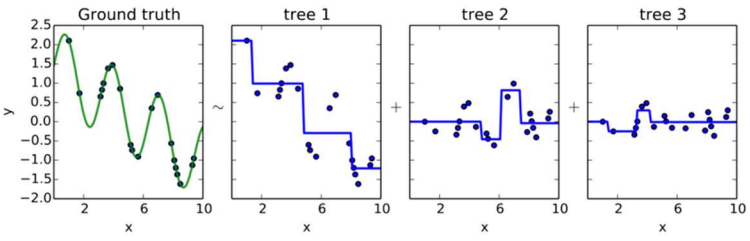
$$Y_M(\mathbf{x}) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right) \qquad \alpha_m = \ln\frac{1-\epsilon_m}{\epsilon_m}$$

이 과정은 지수오류함수의 연속적인 최소화로 해석이 가능합니다. 자세한 과정은 강의에 증명과 함께 다루었습니다. 참고하세요.

adaboost는 구현이 매우 쉽지만 outlier에 민감하고, 출력값을 확률적으로 해석하기 어려운 단점이 있습니다. 그리고 multiclass분류문제에 적용하기도 쉽지 않습니다.

<gradient boost>

gradient boosting은 residual fitting으로 이해하면 쉽습니다. 다음 그림을 보세요



[Peter Prettenhofer, Gradient Boosted Regression Trees. https://github.com/pprett/pydata-gbrt-tutorial/blob/master/slides/slides.pdf]

녹색포인트를 추정하기 위해 gradient boosting(이하 gb)을 쓴 것입니다. 첫번째 트리로 근사한 뒤 그 오차를 모아서 두번째 트리로 다시 '교정'하고 거기서 남은 잔차(residual)을 다시 세번째 트리로 근사한 것입니다.

adaboost와 비슷하게, gb는 트리가 연속됩니다. 트리형태 분류기는 아무것이나 써도 무관합니다.

위의 개략적인 설명을 구체적으로 기술하면,

1. 초기화 $F_0 = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$
2. 모든 잔차에 대해
    a. 잔차계산 $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$
    b. r_im에 대해 트리 학습
    c. 오차계산 $\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$
    d. 함수갱신 $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_m I(x \in R_{jm})$
3. 결과: $F_M(x)$

이렇게 formal하게 써놓으면 잘 와닿지 않지만, 기본개념이나 구현, 계산이 복잡하지 않습니다. 강의에 구체적인 사례를 들어서 예제문제 하나를 풀었으니 참고 바랍니다.

다음시간에는 이번시간에 이어 모델조합(tree)에 관해 계속 배워보겠습니다.

연습문제

1. 지수족 분포는 곱셈연산에 대해 닫혀 있다.
    a. O
    b. 곱셈연산에 대해 닫혀 있기 때문에 사전/사후분포가 계속 같은 형태가 유지되고, 따라서 마르코프 연쇄에서 복잡도를 증가시키지 않는다. 이 성질때문에 선형동적 시스템에서도 사용된다.
2. 칼만필터에서는 HMM에서처럼 backward probability를 계산할 필요가 없다.
    a. O
    b. 앞으로 전진만 하기 때문에 forward probability만 계산한다.
3. 베이지안 모델평균도 모델조합의 일종으로 이해할 수 있다.
    a. X
    b. 베이지안 모델평균은 결국 복잡한 모델 하나와 같으나, 모델조합은 여러 모델의 성질이 뭉뚱그려지는 것이 아니라 혼합계수에 의해 선택되므로 특정 모델로부터의 표본이 그대로 드러난다. 따라서 서로 다른 데이터포인트라면 서로 다른 분포로부터 나올 수 있다.

4. 부트스트랩은 기본적으로 중복을 허용하지 않는다.
    a. X
    b. 특별한 사정이 없는 한 허용한다. 따라서 모집단의 크기보다도 더 큰 샘플집단 여러개를 만들 수도 있다.
5. (여러번 수행한)독립적 모델과 비교했을 때 bagging의 오류는 무조건 줄어든다.
    a. O
    b. 이상적인 경우처럼 (독립모델에 비해) 극단적으로 오류가 줄어들지는 않지만, 충분히 많은 샘플에 대해 실험했을 때 무조건 에러를 줄이는 것을 보일 수 있다.
6. Adaboost에서 weak learner들은 어떤 것을 사용해도 무방하다.
    a. O
    b. 성능만 나와주면 어떤것을 사용해도 무방하다. 랜덤보다 약간 좋은 수준이라도 여러개를 조합해서 성능이 좋은 분류기를 얻을 수 있다는 것이 adaboost의 장점이다.
7. Adaboost는 성능에 비해 구현이 어렵다
    a. X
    b. 간단한 식에 의해 각 학습기의 가중치와, 데이터 포인트의 weight를 정할 수 있으므로, 구현이 간단하다.
8. gradient boost는 순차적으로 여러 tree를 분류잔차에 fitting하는 과정이다.
    a. O
    b. 잔차를 조금씩 더 제거해나간다는 아이디어가 기본.
9. gradient boost를 classification에 쓸 때, regression처럼 F의 출력값을 그대로 쓰면 안된다.
    a. O
    b. 출력이 log odds의 합이기 때문에 확률로 해석할 수 없고, 따라서 0~1로 오도록 변환을 거친다.

	정리하기

1. 선형예측중에서 모델도 시간에 따라 선형으로 변화가능할 수 있는데 그것을 선형동적 시스템이라고 한다
    a. 간단한 모델의 예: 최근것들의 평균
    b. 간단한 모델의 예2: 가장 최근값
2. 선형예측모델은 연쇄의 길이가 길어질수록 복잡도가 증가하면 쓸수가 없다.(연쇄가 무한할 때 가정)
3. 칼만필터
    a. 선형예측시스템

b. 잠재변수가 마르코프연쇄를 이룸
c. 잠재변수, 노이즈 모두 가우시안을 가정한다.
d. EM으로 학습 및 예측한다

4. 베이지안 모델평균과 모델조합은 다르다.
   a. 모델조합은 데이터집합의 각 포인트가 서로 다른 잠재변수에 기반할수 있음
   b. 베이지안 모델평균은 모델이 결국 하나다.
5. bootstrap: 데이터집합에서 특정수의 샘플을 뽑아서 새로운 데이터셋을 만드는 것.
   데이터포인트의 중복을 허용한다.
6. bagging : bootstrap aggregating
   a. bootstrap으로 여러 데이터셋을 만들고 각 데이터셋에 대해 학습한 모델의 조합으로 예측
   b. 개별적으로 M개의 모델을 시험하면 평균오류는 1/M
   c. 하지만 bagging에서 오류는 서로 높은 상관관계를 가지게 되므로 오류가 이렇게 낮지는 않다.
   d. 그래도 오류가 무조건 줄어든다는 것이 알려져 있다.
7. adaboost
   a. adaptive boosting
   b. weak learner여러개를 모아서 좋은 분류기를 만들 수 있다.
   c. weak learner들은 순차적으로 학습되며, 앞단의 분류기 결과에 따라 데이터 포인트의 weight가 재조정된다.
   d. '지수오류함수의 연속적인 최소화'로 해석이 가능하다.
   e. multiclass문제에 적용하기가 까다롭다.
8. gradient boosting
   a. 기본적으로 regression
   b. residual fitting으로 이해할 수 있다.
   c. leaf가 8~32개인 tree의 연속.
   d. classification에도 적용 가능하다.

	참고하기

Bishop, C. M. "Bishop-Pattern Recognition and Machine Learning-Springer 2006." Antimicrob. Agents Chemother (2014): 03728-14.

	다음 차시 예고

- 트리기반 모델조합
    - decision tree
    - 선형회귀의 혼합
- 확률분포
    - 베르누이분포
    - 이항분포
    - 베타분포