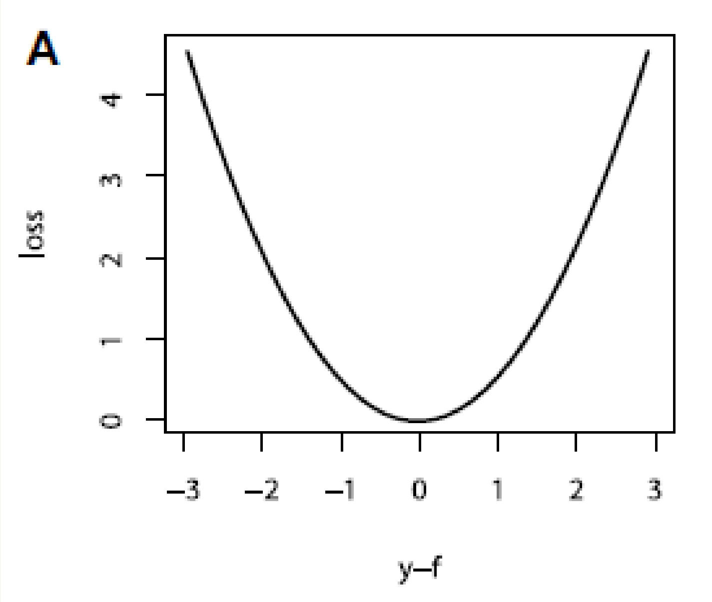
Gradient Boosting Machines

1. GBM Design

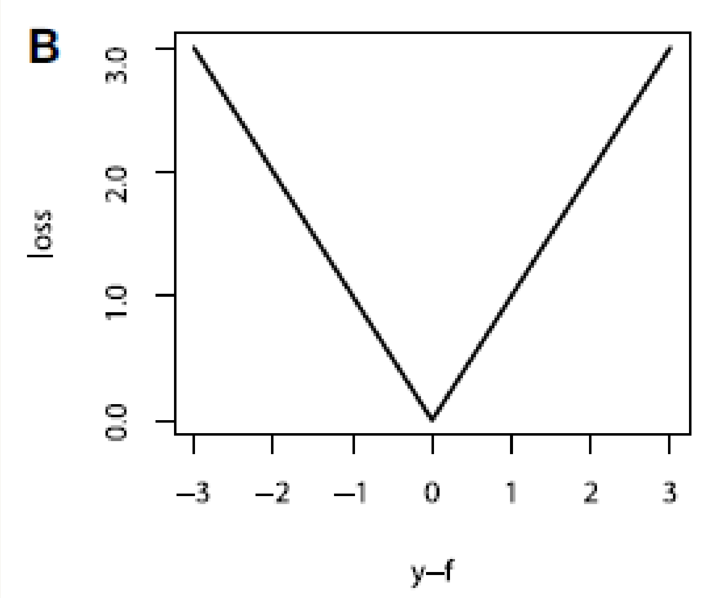
- **Continuous response, $y \in R$:**
  - Gaussian L2 loss function
  - Laplace L1 loss function
  - Huber loss function, δ specified
  - Quantile loss function, α specified
- **Categorical response, $y \in \{0, 1\}$:**
  - Binomial loss function
  - Adaboost loss function
- **Other families of response variable:**
  - Loss functions for survival models
  - Loss functions counts data
  - Custom loss functions

1.1 Loss functions for continuous response

- A. L2 squared loss function



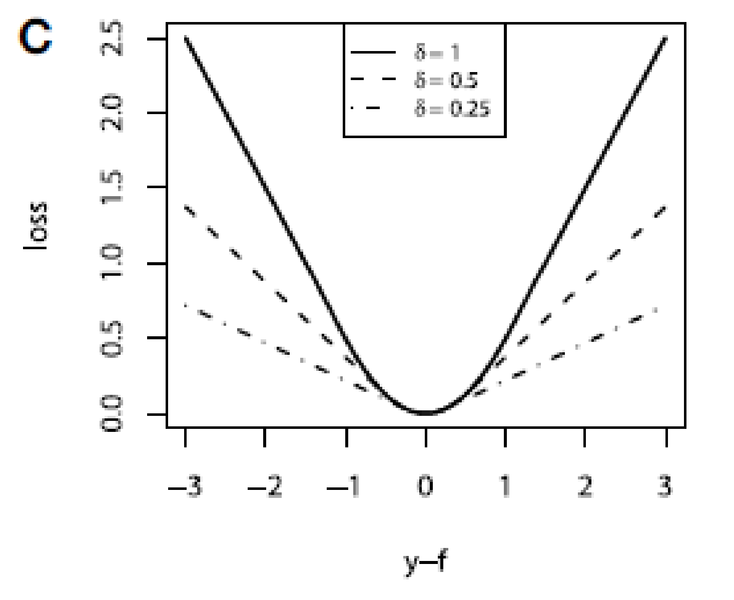
- B. L1 absolute loss function



$$\Psi(y,f)_{L_1} = |y - f|$$

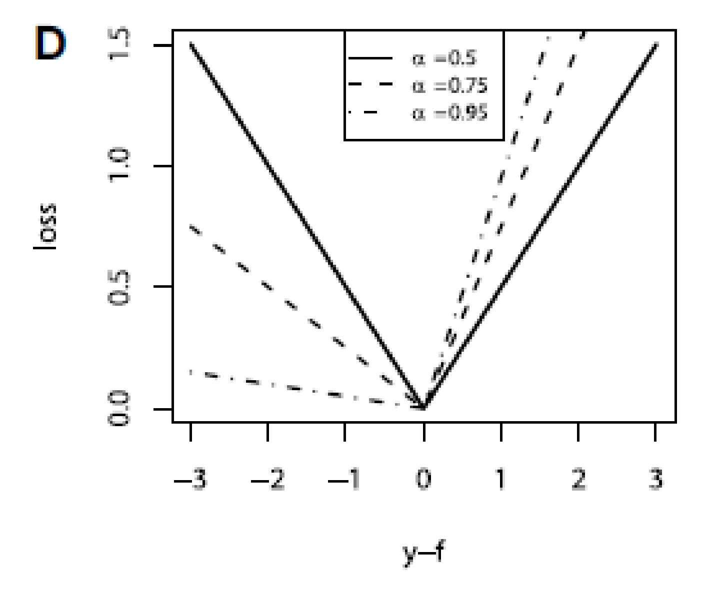
1.1 Loss functions for continuous response

- **C. Huber loss function**



$$\Psi(y,f)_{\text{Huber},\,\delta} = \begin{cases} \frac{1}{2}(y-f)^2 & |y-f| \leq \delta \\ \delta(|y-f| - \delta/2) & |y-f| > \delta \end{cases}$$

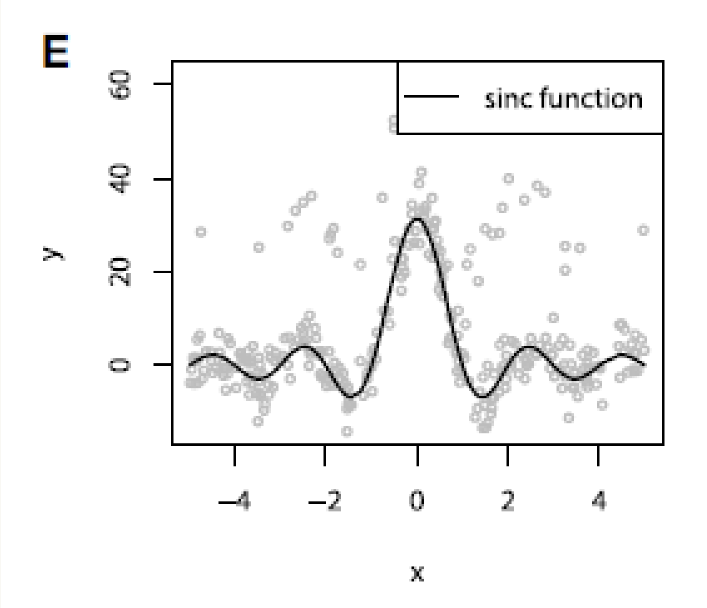
- **D. Quantile loss function**



$$\Psi(y,f)_{\alpha} = \begin{cases} (1-\alpha)|y-f| & y-f \leq 0 \\ \alpha|y-f| & y-f > 0 \end{cases}$$

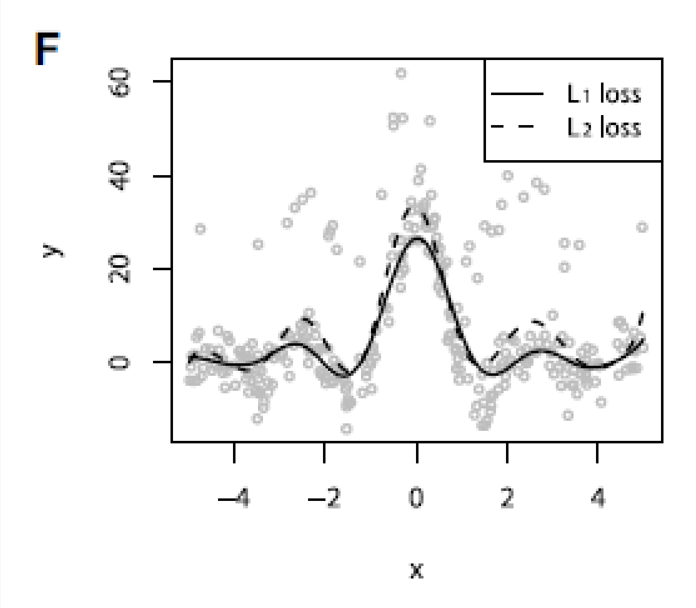
1.1 Loss functions for continuous response

- E. Original sinc(x) function



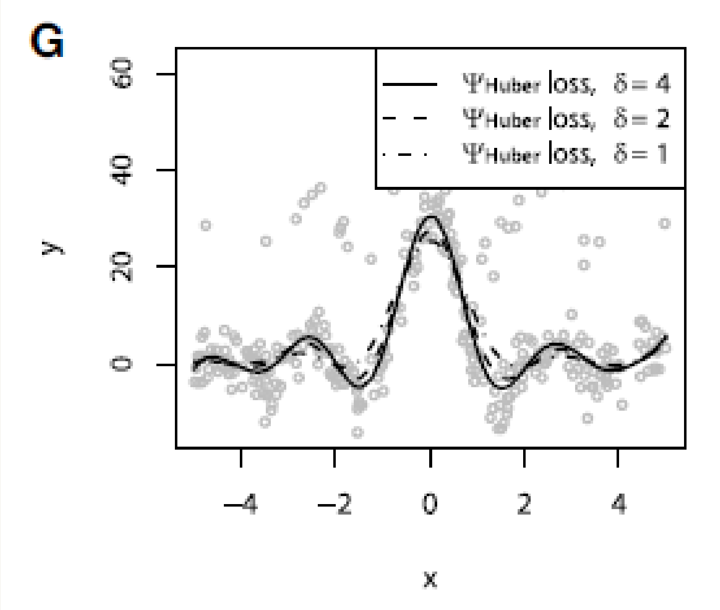
$$* \; sinc(x) = \frac{\sin(\pi x)}{\pi x}$$

- F. smooth GBM fitted with L2 and L1 loss

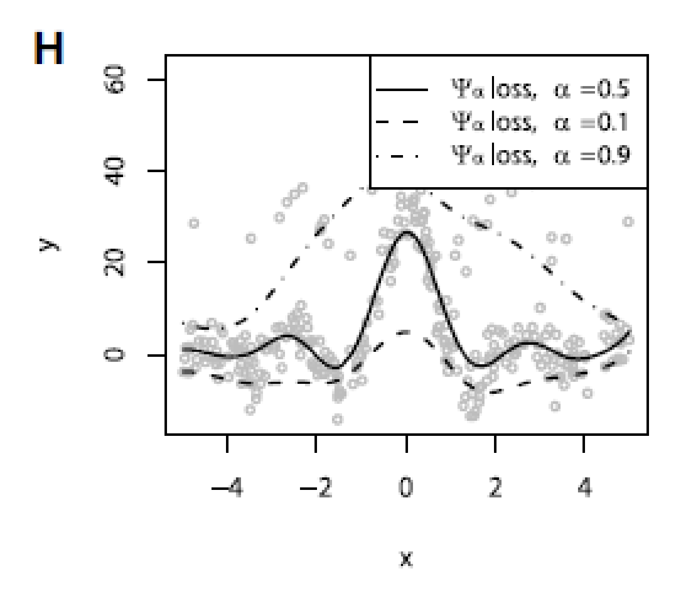


1.1 Loss functions for continuous response

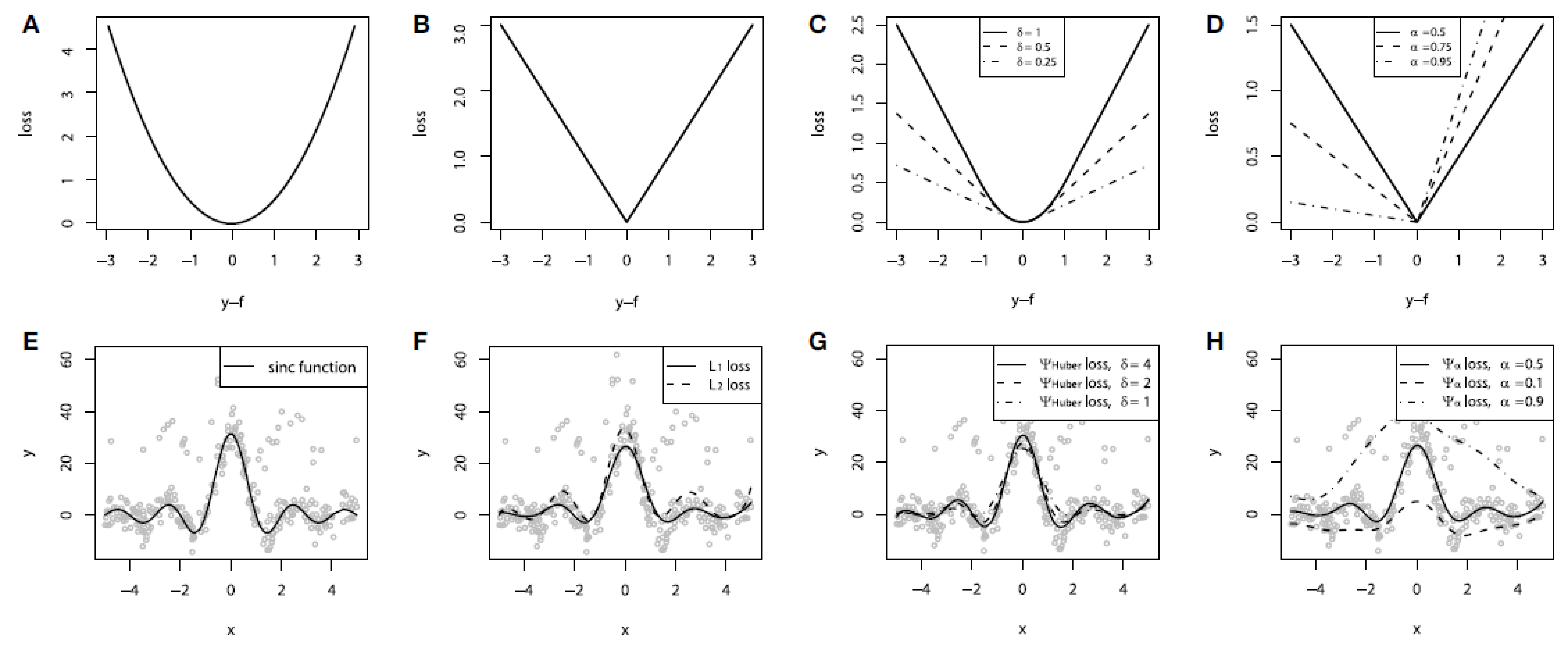
- G. smooth GBM fitted with Huber loss with δ = 4, 2, 1



- H. smooth GBM fitted with Quantile loss with α = 0.5, 0.1, 0.9

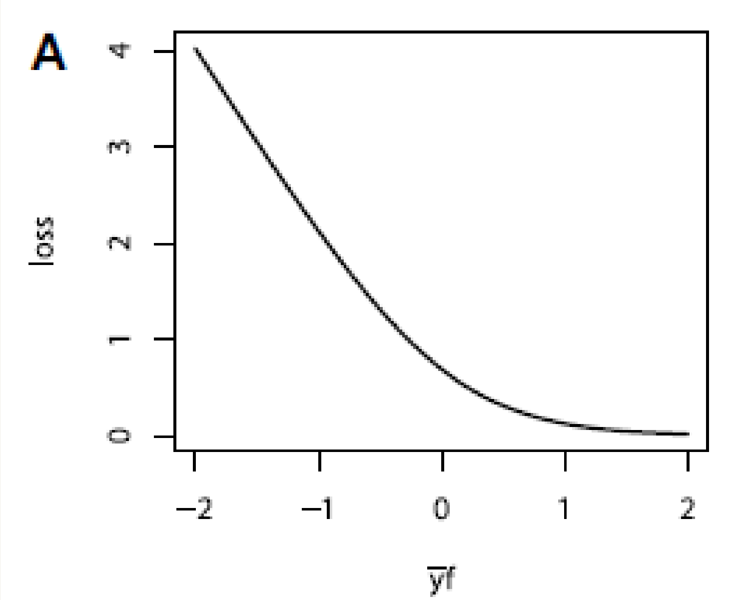


1.1 Loss functions for continuous response



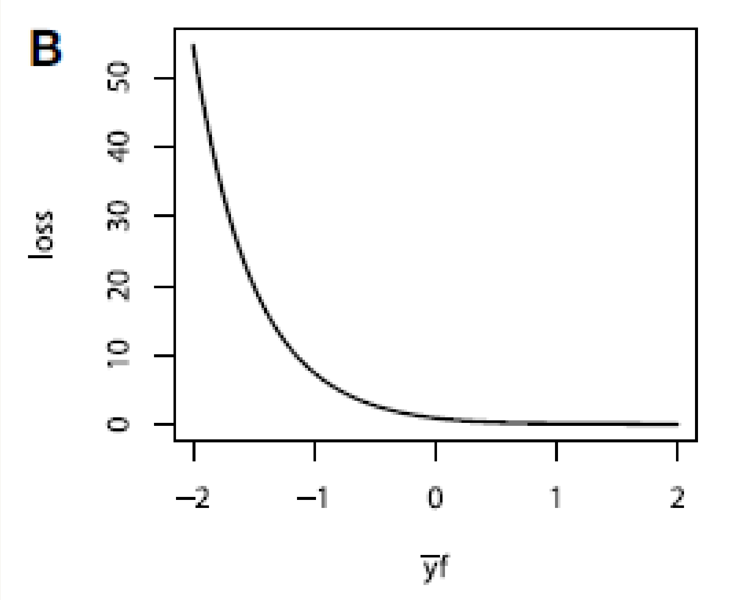
1.2 Loss functions for categorical response

- A. Bernoulli loss function



$$\Psi(y, f)_{\text{Bern}} = \log(1 + \exp(-2\bar{y}f))$$

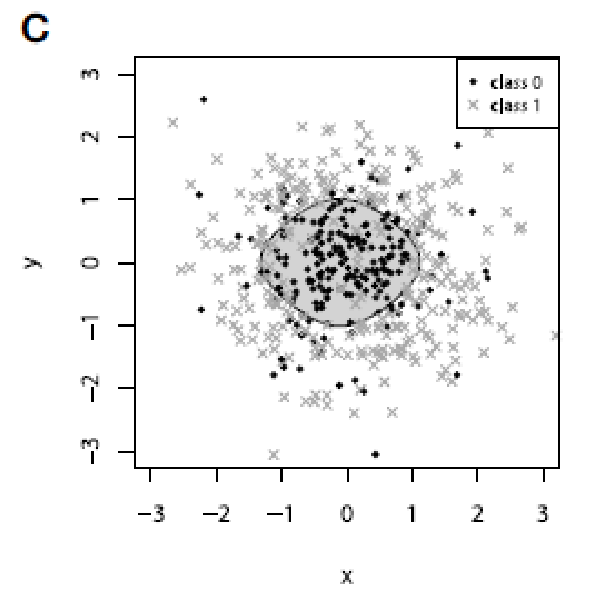
- B. Adaboost loss function



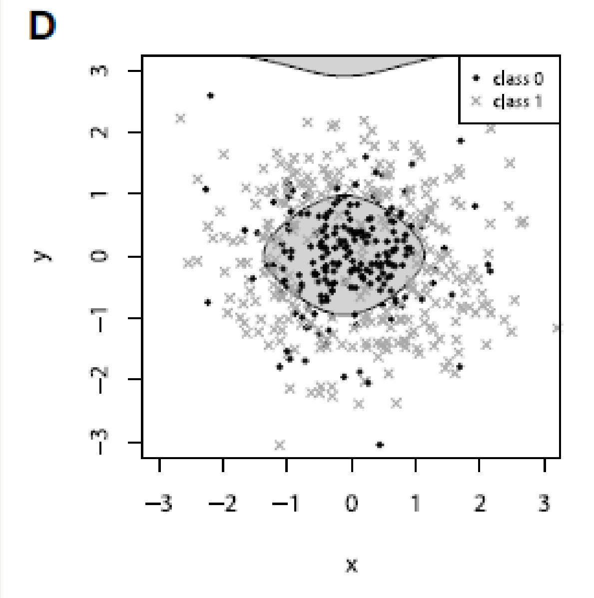
$$\Psi(y, f)_{\text{Ada}} = \exp(-\bar{y}f)$$

1.2 Loss functions for categorical response

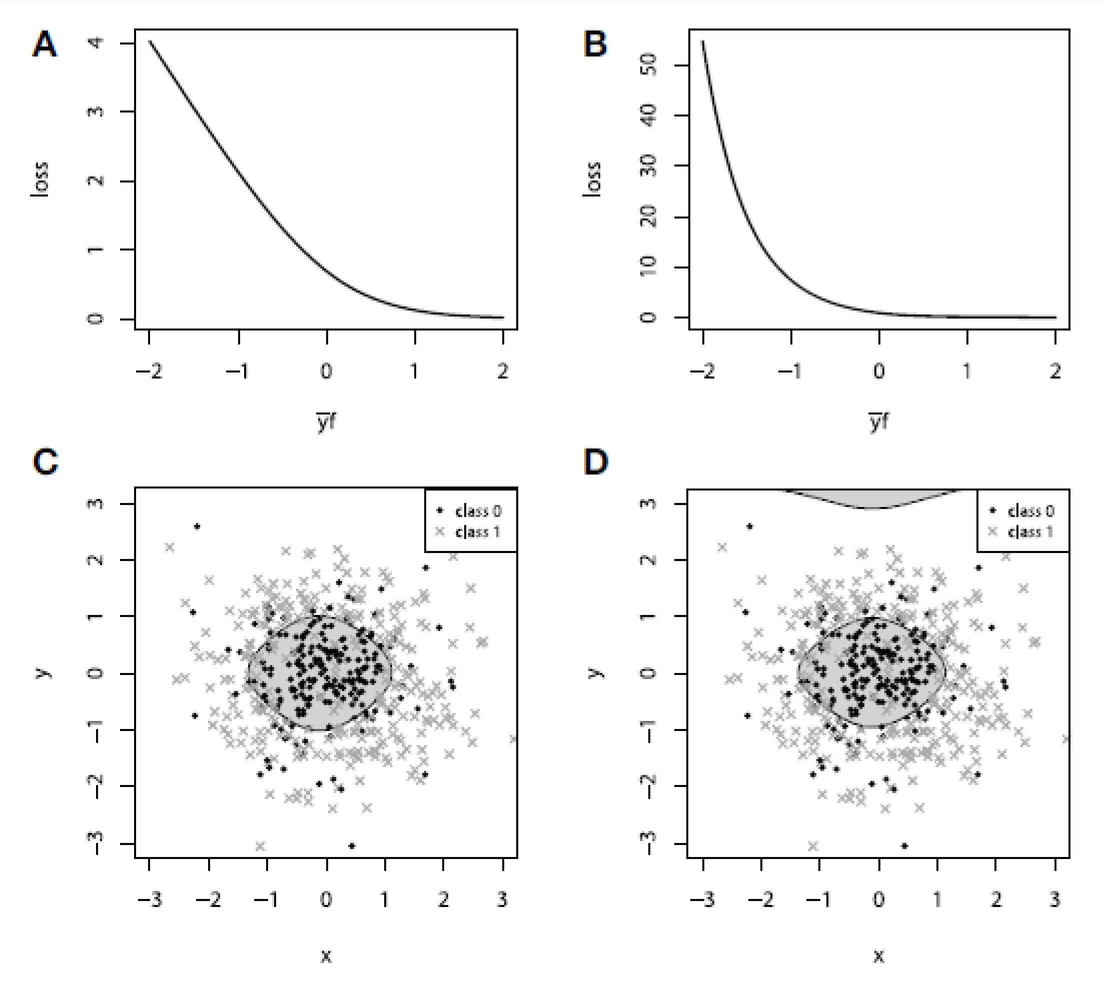
- C. GBM 2D classification with Bernoulli loss



- D. GBM 2D classification with Adaboost loss

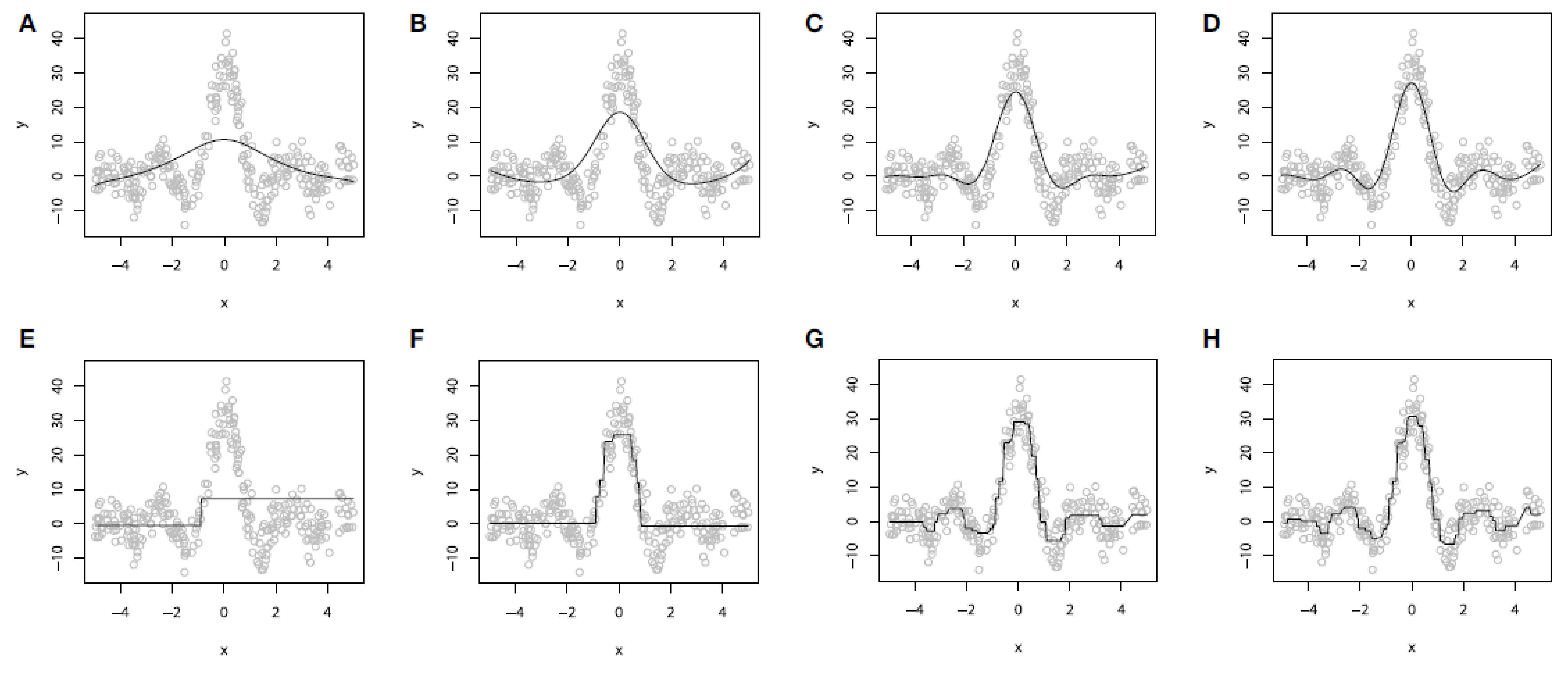


1.2 Loss functions for categorical response



1.3 Base - learner models

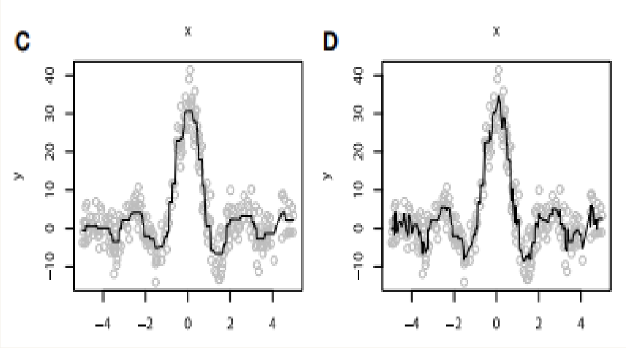
- **Linear models:**
  - Ordinary linear regression
  - Ridge penalized linear regression
  - Random effects
- **Decision trees**
  - Decision tree stumps
  - Decision trees with arbitrary interaction depth
- **Smooth models:**
  - P-splines
  - Radial basis functions
- **Othermodels:**
  - Markov Random Fields
  - Wavelets
  - Custom base-learner functions

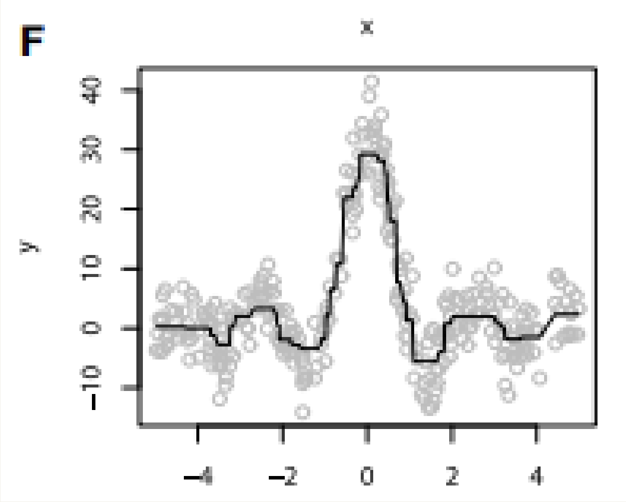


- **A~D: P-Spline GBM model (M = 1, 10, 50, 100)**
- **E~H: Decision Tree GBM model (M = 1, 10, 50, 100)**

2. Overfitting REGULARIZATION

< 수축 정규화를 이용한 예 >





2.1 subsampling

iteration을 하는 순간에 기존 데이터의 일부분만 가지고 random sampling (복원, 비복원 둘 다 가능)을 진행한 후, 모델을 학습시킨다.
-> 일반화 성능의 향상

2.2 SHRINKAGE

-> 모델 복잡성을 제어하는 고전적인 접근법은 수축을 통한 정규화의 도입

-> GBM의 맥락에서 수축은 각 iteration에 따른 모델의 영향을 줄이거나 축소하는 데 사용됨

-> 수축에 의한 정규화의 가장 간단한 형태는 직접 비례 수축이다.

-> 이러한 규제는 GBM 알고리즘의 마지막 단계에 적용된다.

$$\widehat{f_t} \leftarrow \widehat{f}_{t-1} + \lambda \rho_t h(x, \theta_t)$$

-> 매개 변수 λ가 작을수록, 축소된 부스트 증분이 낮을수록 일반화가 더 잘 이루어지는 것이 일반적이다.

* 여기서 파라미터 즉 사용자가 조절하는 것은 λ 에 해당.

2. Overfitting REGULARIZATION

2.3 early stopping

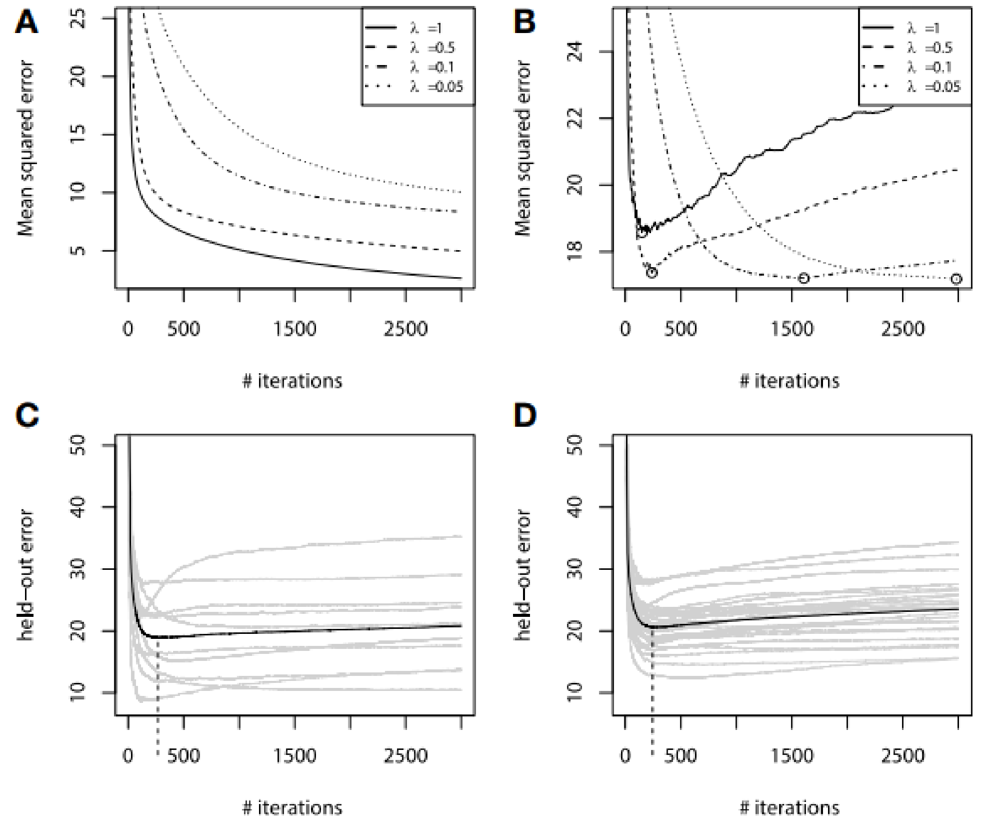


FIGURE 5 | Error curves for GBM fitting on sinc(x) data: (A) training set error; (B) validation set error. Error curves for learning simulations and number of base-learners *M* estimation: **(C)** error curves for cross-validation; **(D)** error curves for bootstrap estimates.

training data set
iteration(M) ↑, error ↓

validation data set
iteration(M) 일정 수준 ↑, error ↑
(∵overfitting)

∴ early stopping을 통해
overfitting regularization
(딥러닝에서도 응용)

Thank you