비타민 8주차 세션

비타민 10기 6조
권동구 김은비 조성우 조형주

8주차 세션 진행

복습세션

회귀 복습

서포트 벡터 머신

실습

7주차 복습과제 리뷰

0. 데이터 불러오기

features_cat 데이터 숫자로 인코딩

```
features_cat['Sex'] = pd.get_dummies(features_cat['Sex'])
features_cat['Embarked'] = pd.get_dummies(features_cat['Embarked'])
features_cat['Pclass'] = pd.get_dummies(features_cat['Pclass'])
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call
last)
<ipython-input-13-d6b4b6389847> in <module>
----> 1 features_cat['Sex'] = pd.get_dummies(features_cat['Sex'])
      2 features_cat['Embarked'] =
pd.get_dummies(features_cat['Embarked'])
      3 features_cat['Pclass'] = pd.get_dummies(features_cat['Pclass'])

1 frames

/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py in
_set_item_frame_value(self, key, value)
   3727             len_cols = 1 if is_scalar(cols) else len(cols)
   3728             if len_cols != len(value.columns):
-> 3729                 raise ValueError("Columns must be same length as
key")
   3730
   3731             # align right-hand-side columns if self.columns

ValueError: Columns must be same length as key
```

SEARCH STACK OVERFLOW

레이블 인코딩

```
[17] from sklearn.preprocessing import LabelEncoder
     le = LabelEncoder()
```

```
features_cat['Sex'] = le.fit_transform(features.Sex.values)
features_cat['Embarked'] = le.fit_transform(features.Embarked.values)
features_cat['Pclass'] = le.fit_transform(features.Pclass.values)
```

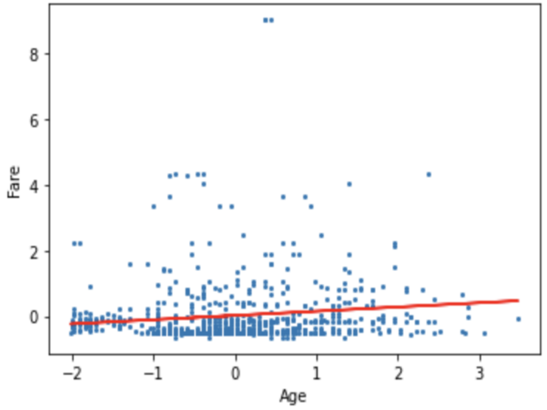
```
features_cat['Sex'] = features_cat['Sex'].map({'male':0, 'female':1}).astype('int')
features_cat['Embarked'] = features_cat['Embarked'].map({'S':0,'C':1,'Q':2})
features_cat['Pclass'] = features_cat['Pclass'].astype('int')
```

1. 경사하강법

1) 배치경사하강법(BGD)

```
[ ] plt.scatter(df['Age'],df['Fare'],s = 4)
    plt.plot(df['Age'],w*np.array(df['Fare']) + b,c = 'red')
    plt.xlabel('Age')
    plt.ylabel('Fare');
```

```
[ ] plt.scatter(features_con_std['Age'], features_con_std['Fare'], s=4)
    plt.plot(features_con_std['Age'], w*np.array(features_con_std['Age'])+b, c='red')
    plt.xlabel('Age')
    plt.ylabel('Fare')
```



→ 표준화된 값 넣기

2) 확률적 경사하강법

```
[ ]  1 mean_squared_error(y_pred, y_test)

1.7357651391990743
```

→ (예측값, 실제값) 순서 유의

3) 미니배치 경사하강법

2. 손실함수

1) Cross-Entropy

cross-entropy와 정확도를 봅시다

```
[ ]  1 cross_entropy(pred, np.array(y_test))
```

46.926110280282145

```
[ ]  1 accuracy_score(np.round(pred).astype('int'),np.array(y_test))
```

0.822429906542056

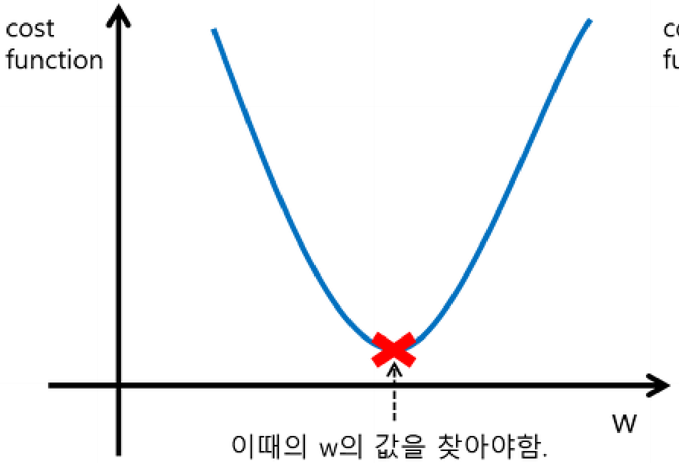
→ pred값 넣을 때 integer 형식으로 설정

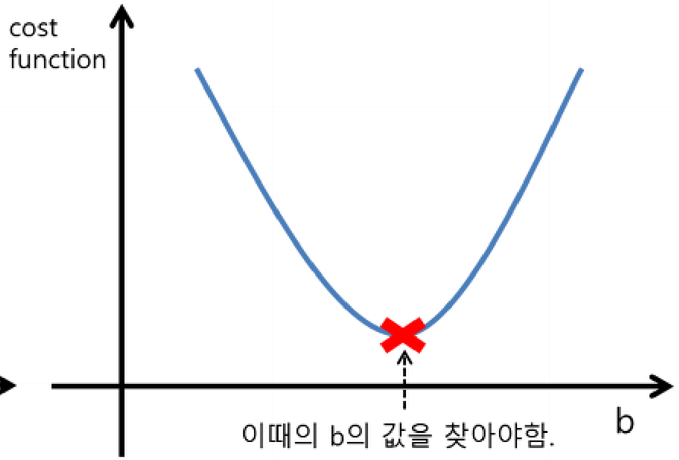
7주차 복습

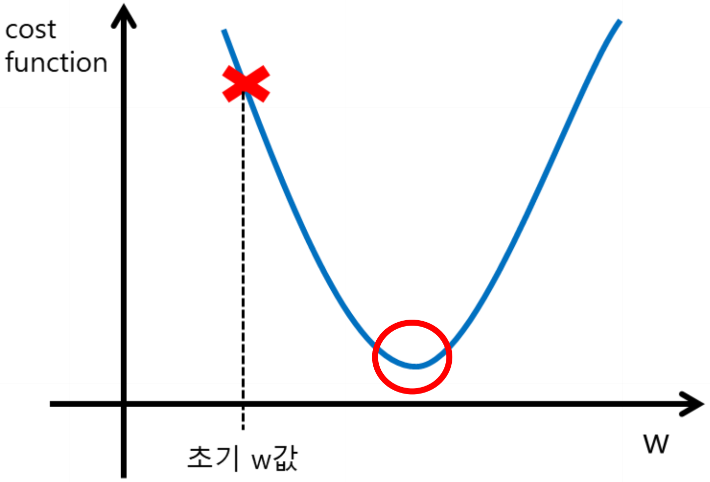
$$\hat{y_i} = wx_i + b$$

$MSE = \sum(y_i - \hat{y_i})$ 를 최소로 만드는 w, b를 찾는 방법

경사하강법







$$w_{new} = w - \alpha \frac{\partial}{\partial w} MSE$$

학습률 parameter α

경사하강법

01. 확률적 경사하강법

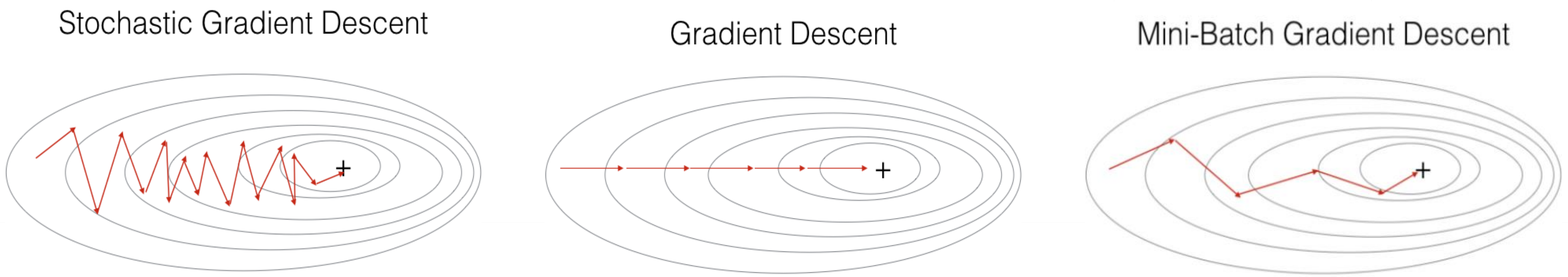
훈련 세트에서 랜덤하게 하나의 샘플을 골라 경사하강법 진행
→ 전체 샘플을 모두 이용할 때까지

02. 배치 경사하강법

전체 데이터 샘플을 한번에 사용
→ 안정적이지만 local minimum에 수렴하면 탈출이 어려움

03. 미니배치 경사하강법

배치경사하강법 & 확률적 경사하강법의 절충안
→ 훈련세트에서 몇 개의 샘플을 선택



손실함수 (비용함수)

01. 회귀　　MSE, RMSE, MAE

02. 분류　　Cross Entropy

$$logloss = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log(p_{ij})$$

N is the number of rows
M is the number of classes

(M=2) 이진 분류 – Binary Cross Entropy

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log (1 - \hat{Y}_i))$$

	Actual (Y_i)	Predicted Probability ($\hat{Y}_i$)	Log($\hat{Y}_i$)
1	1	0.9	−0.1053605
2	1	0.7	−0.3566749
3	0	0.3 (0.7)	−1.203973
4	0	0.5 (0.5)	−0.6931472

$$-\{(1 * (-0.1053605) + 0) + (1 * (-0.3566749) + 0) + (0 + 1 * (-0.3566749) + (0 + 1 * (-0.6931472)\}/4 = 0.3779644$$

Cross Entropy – Multiclass classification

Loss function	Usage	Examples: Using probabilities (from_logits=False)	Examples: Using logits (from_logits=True)
BinaryCrossentropy	Binary classification	y_true: 1 y_pred: 0.69	y_true: 1 y_pred: 0.8
CategoricalCrossentropy	Multiclass classification	y_true: 0 0 1 y_pred: 0.30 0.15 0.55	y_true: 0 0 1 y_pred: 1.5 0.8 2.1
Sparse CategoricalCrossentropy	Multiclass classification	y_true: 2 y_pred: 0.30 0.15 0.55	y_true: 2 y_pred: 1.5 0.8 2.1

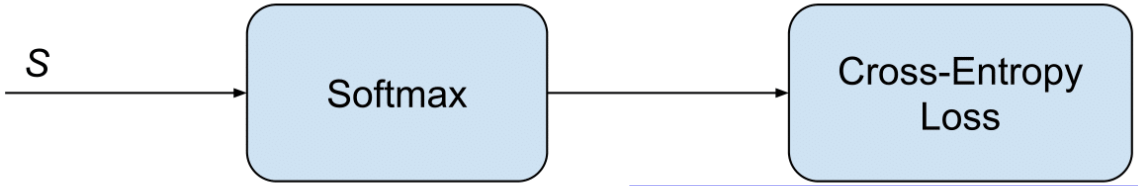
1. Categorical Cross Entropy

$$logloss = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log(p_{ij})$$

N is the number of rows
M is the number of classes

(M ≥ 3)
분류해야 할 클래스가 3개 이상인 경우

☞ 데이터 label이 [0,0,1,0,0], [1,0,0,0,0], [0,0,0,1,0]처럼 원-핫 인코딩 방식일 때 사용



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$$

$$\boxed{CE = -\sum_{i}^{C} t_i log(f(s)_i)}$$

일반적인 Softmax Loss

$$L = -\frac{1}{N}\sum_{j=1}^{N}\boxed{\sum_{i=1}^{C} t_{ij} log(y_{ij})}$$

N: $observation$ 개수 / C: $class$ 개수
t_{ij}: 실제값
y_{ij}: 예측값 / 관측값

1. Categorical Cross Entropy

ex)

obs	Target Feature
1	TV
2	Phone
3	Camera
4	TV

→

TV	Phone	Camera
1	0	0
0	1	0
0	0	1
1	0	0

One-hot encoding한 실제값

TV	Phone	Camera
0.6	0.1	0.3

⋮

1번의 관측값 / 예측값

$$L = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{C} t_{ij} log(y_{ij})$$

$$\begin{aligned} L &= -(1 * \log(0.6) + 0 * \log(0.1) + 0 * \log(0.3)) \\ &= -\log(0.6) \approx 0.5108256 \end{aligned}$$

2. Sparse Categorical Cross Entropy

$$logloss = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log(p_{ij})$$

N is the number of rows
M is the number of classes

→ (M ≥ 3)
분류해야 할 클래스가 3개 이상인 경우

☞ 데이터 label이 [1,2,3,4]처럼 레이블 인코딩 방식 (정수 인코딩 상태)일 때 사용

Cross entropy loss 함수는 동일
정수 label을 원-핫 인코딩 형식으로 바꿔주기만 하면 됨

$$L = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{C} t_{ij}log(y_{ij})$$

2. [실습코드] Categorical Cross Entropy

▷ keras에서 지원하는 범주형 교차 엔트로피 (딥러닝)

```
tf.keras.losses.BinaryCrossentropy( from_logits=False,
label_smoothing=0, reduction="auto",
name="binary_crossentropy" )

tf.keras.losses.CategoricalCrossentropy(from_logits=Fa
lse, label_smoothing = 0, reduction="auto", name =
‘categorical_crossentropy")

sparse_categorical_loss =
tf.keras.losses.SparseCategoricalCrossentropy()
sparse_categorical_loss(y_test, y_pred).numpy()
```

```
[ ]
import tensorflow as tf

tf.keras.losses.BinaryCrossentropy(
    from_logits=False, label_smoothing=0, reduction='auto',
    name='binary_crossentropy'
)

tf.keras.losses.binary_crossentropy(
    y_test, y_pred, from_logits=False, label_smoothing=0
)
```

3. A General and Adaptive Robust Loss Function

Loss를 최소화하는 것이 목표

→ 어떤 loss function을 써야 가장 좋은 결과(parameter 예측)를 도출할지 결정하기 어려울 수 있음

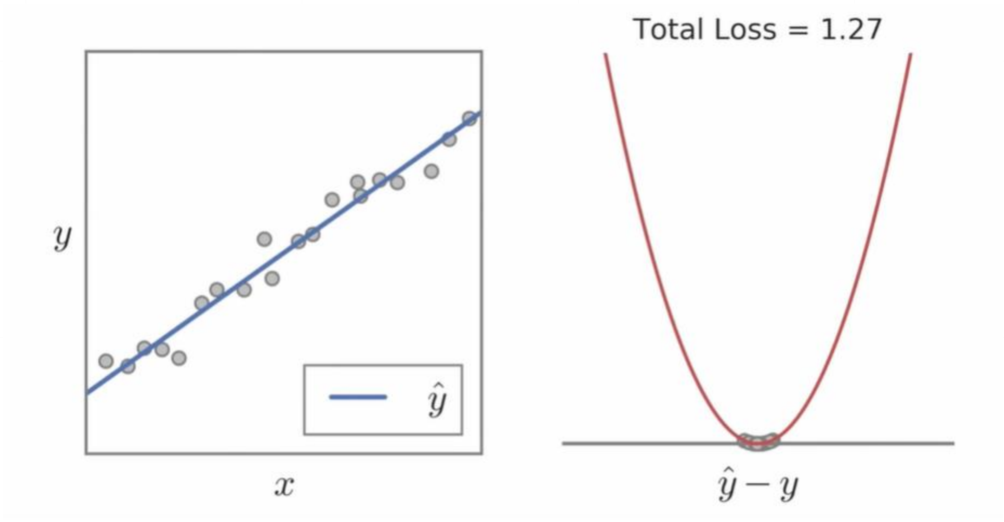
1) General family of loss functions — variable robustness

2) Adaptive loss function — 1)에서 나온 function 중 어떤 것을 써야 하는지
Determine robustness

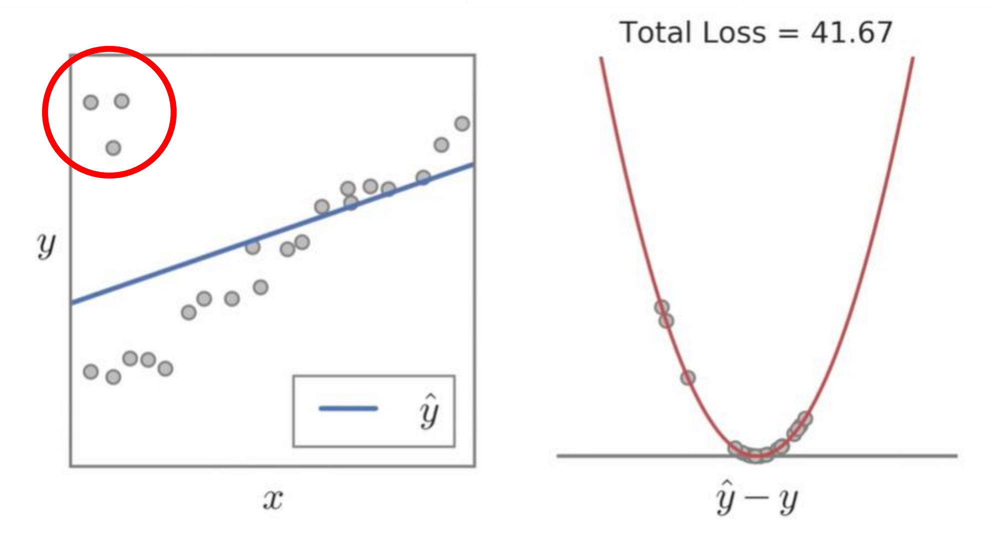
즉, outlier에 영향을 덜 받는 loss function을 찾는 것

https://www.youtube.com/watch?v=BmNKbnF69eY

3. A General and Adaptive Robust Loss Function

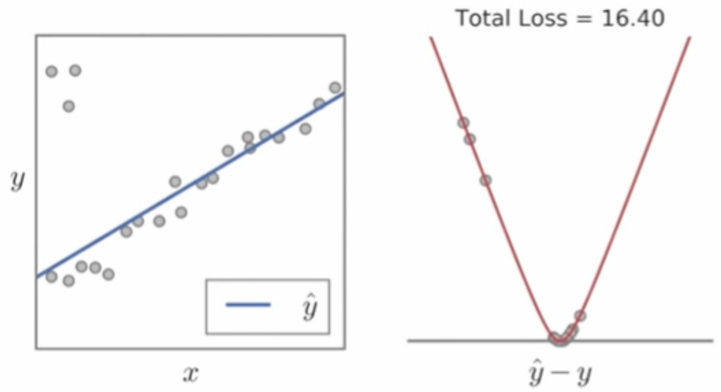


L_2

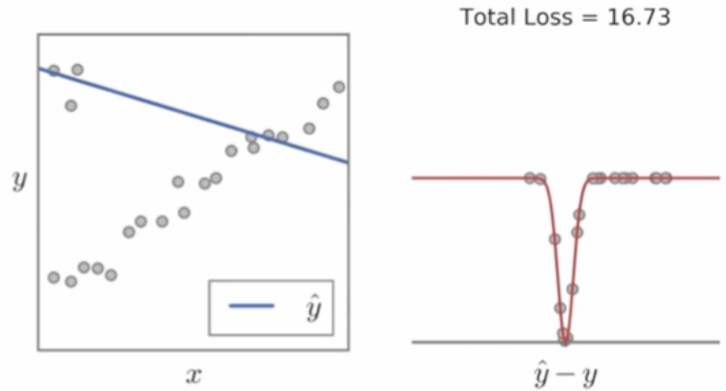


L_2

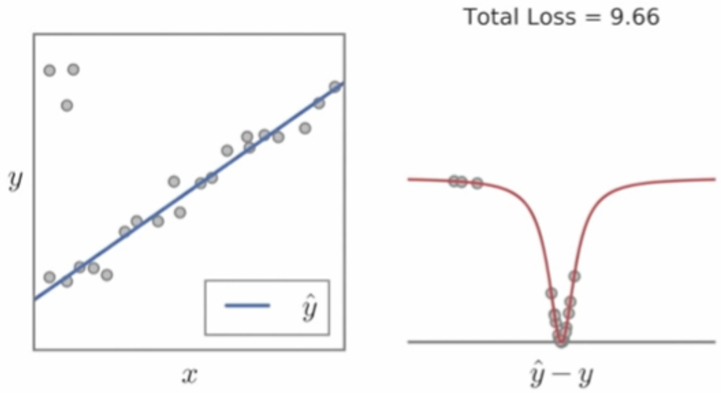
3. A General and Adaptive Robust Loss Function



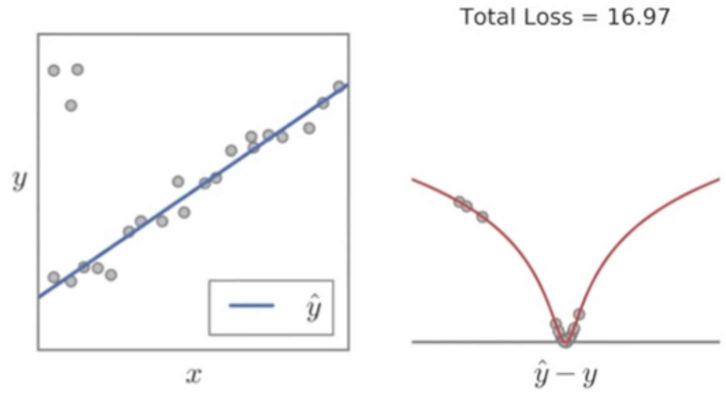
Pseudo-Huber (a.k.a. Charbonnier, Smooth L_1)
Huber, 1964
Charbonnier et al., 1994



Welsch (a.k.a. Leclerc)
Dennis & Welsch,1978
Leclerc, 1989

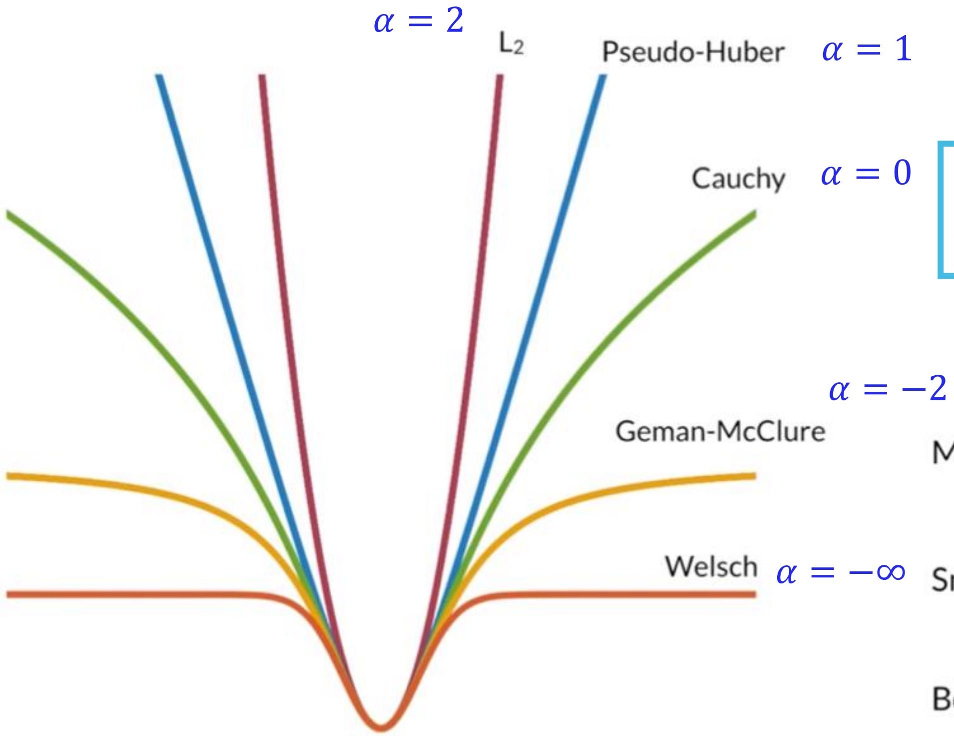


Geman-McClure
Geman & McClure, 1985



Cauchy (a.k.a. Lorentzian)
Poisson,1824
Black & Anandan, 1996

3. A General and Adaptive Robust Loss Function



$$\boxed{\rho(x,\alpha) = \frac{|2-\alpha|}{\alpha}\left(\left(\frac{x^2}{|2-\alpha|}+1\right)^{\alpha/2} - 1\right)}$$

Monotonic w.r.t. $|x|$ and α
(useful for graduated non-convexity)

Smooth w.r.t. x and α
(i.e., in C^∞)

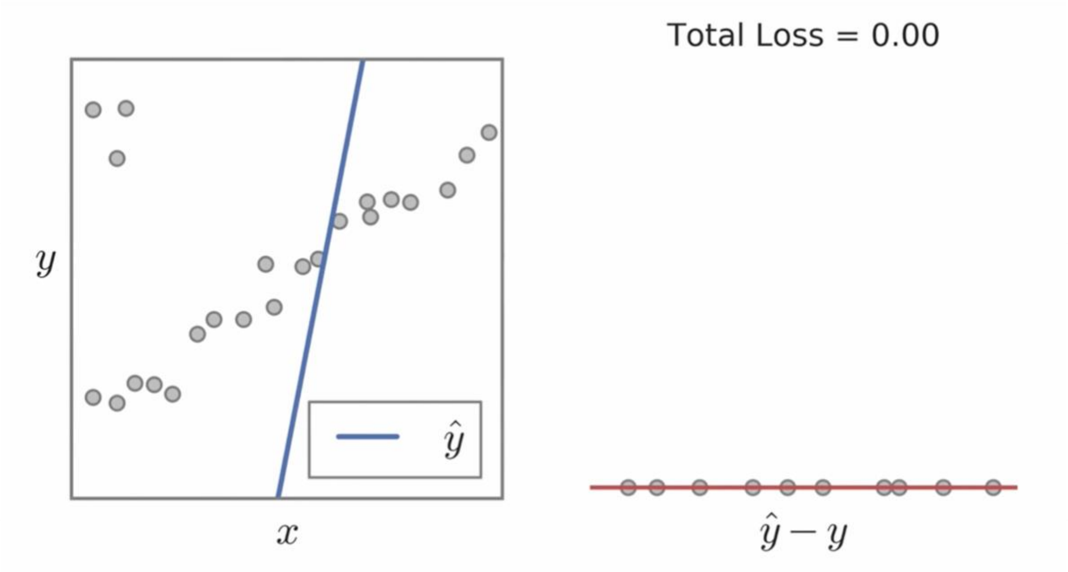
Bounded first and second derivatives
(no exploding gradients, easy preconditioning)

3. A General and Adaptive Robust Loss Function

alpha를 미리 지정하지 않고 free parameter로 두면…

⇒ 따라서 p(x,a)를 minimize하는 것이 아니라

$$\boxed{-\log p(x|\alpha) = \rho(x,\alpha) + \log Z(\alpha)}$$



$$Z\left(\frac{n}{d}\right) = \frac{e^{\left|\frac{2d}{n}-1\right|}\sqrt{\left|\frac{2d}{n}-1\right|}}{(2\pi)^{(d-1)}} G^{0,0}_{p,q}\left(\begin{matrix}\mathbf{a_p}\\ \mathbf{b_q}\end{matrix} \,\middle|\, \left(\frac{1}{n}-\frac{1}{2d}\right)^{2d}\right)$$

$$\mathbf{b_q} = \left\{\frac{i}{n}\,\middle|\, i = -\frac{1}{2}, \ldots, n-\frac{3}{2}\right\} \cup \left\{\frac{i}{2d}\,\middle|\, i = 1, \ldots, 2d-1\right\}$$

$$\mathbf{a_p} = \left\{\frac{i}{n}\,\middle|\, i = 1, \ldots, n-1\right\}$$

이렇게 할 경우 alpha를 free parameter로 두어도 모델을 train하는 과정에서 적절한 alpha값을 찾아줌

감사합니다