Applied Machine Learning

Robust Linear Regression

Robust Linear Regression

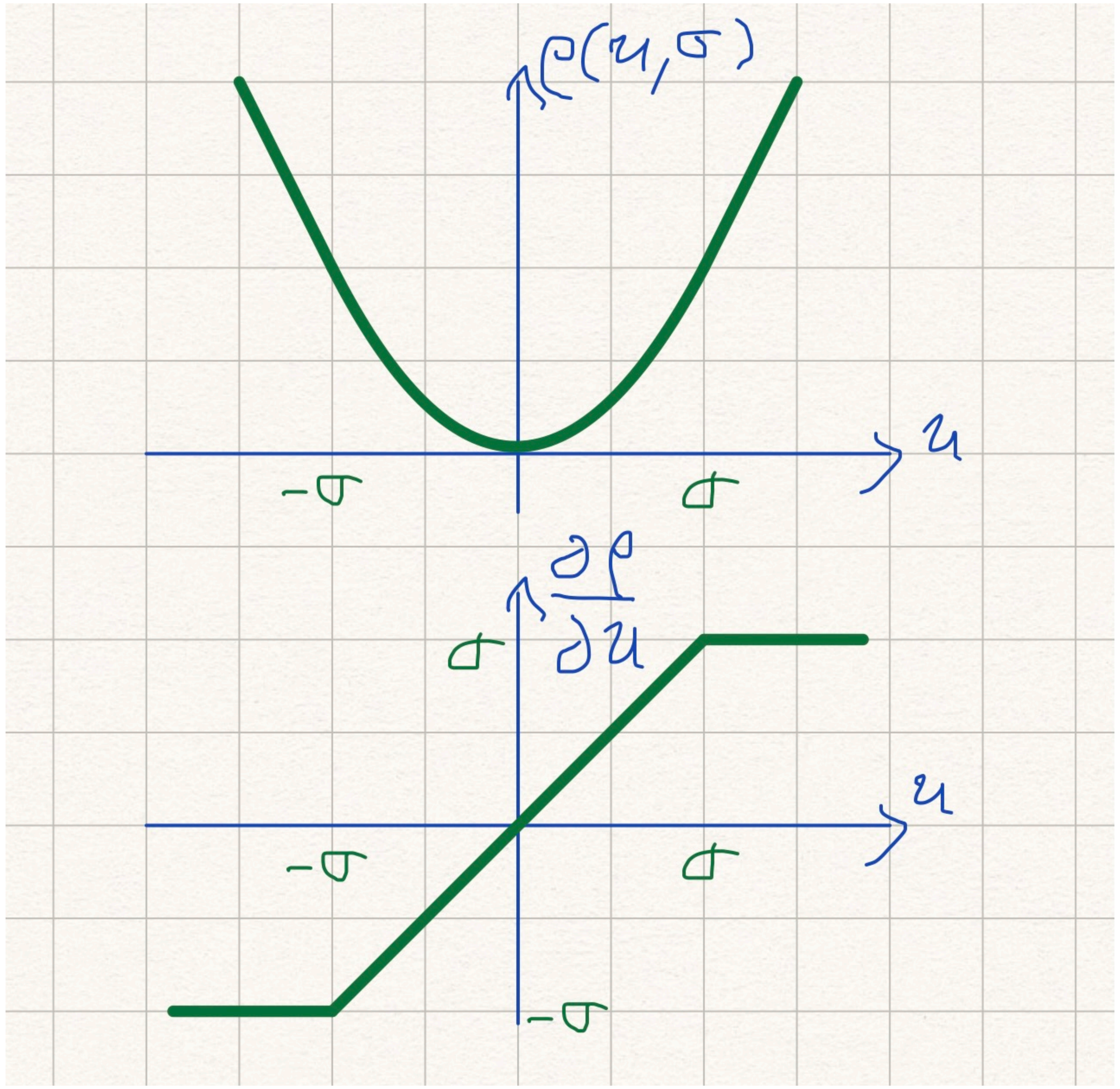
- M-Estimators
- Minimization function with M-Estimators
- Solving regression with M-Estimators

Robust Linear Regression

- Regression finds coefficients that minimize $\sum_i e^2$
    - Single point with large residual pull regression towards it
    - likely outliers
- Weight points in cost function
    - large for those with low residuals
    - low for those with high residuals

M-Estimators

- Linear regression minimizes: $\sum_i (r_i(\mathbf{x}_i, \beta))^2$
  - Model coefficients: β
  - Residual for i'th point: $r_i(\mathbf{x}_i, \beta) = y_i - \mathbf{x}_i^\top \beta$
- M-estimators minimize: $\sum_i \rho(u, \sigma)$
  - Hubber loss: $\rho(u, \sigma) = \begin{cases} \frac{u^2}{2} & |u| < \sigma \\ \sigma|u| - \frac{\sigma^2}{2} & |u| \geq \sigma \end{cases}$
  - $u = r_i(\mathbf{x}_i, \beta)$



Influence Function of an M-Estimator

- $\dfrac{\partial \rho}{\partial u}$
- Weight for point i: $$w_i(\beta) = \frac{\frac{\partial \rho}{\partial u}}{y_i - \mathbf{x}_i^\top \beta}$$
- Minimization criterion: $$\begin{aligned} \nabla_\beta \Big(\textstyle\sum_i \rho(y_i - \mathbf{x}_i^\top \beta, \sigma) \Big) &= \textstyle\sum_i \left[\frac{\partial \rho}{\partial u}\right](-\mathbf{x}_i) &= 0 \\ &= \textstyle\sum_i (w_i(\beta)(y_i - \mathbf{x}_i^\top \beta)(-\mathbf{x}_i) &= 0 \end{aligned}$$
- Minimization in matrix form:
  - $\mathbf{W}(\beta)$: diagonal matrix with $w_{i,i} = w_i(\beta)$
  - $\mathbf{X}^\top[\mathbf{W}(\beta)]\mathbf{y} = \mathbf{X}^\top[\mathbf{W}(\beta)]\mathbf{X}\beta$

Solving regression with M-Estimator

- minimization criterion: $\mathbf{X}^\top[\mathbf{W}(\beta)]\mathbf{y} = \mathbf{X}^\top[\mathbf{W}(\beta)]\mathbf{X}\beta$
  - Initialization: fit $\hat{\beta}^{(1)}$ to a randomly sampled small subset of training set
  - while changes in $W(\beta)$ are higher than threshold at step (n)
    - update elements in $W^{(n)}$ with: $w_i^{(n)} = \dfrac{\frac{\partial \rho}{\partial u}}{y_i - \mathbf{x}_i^\top \hat{\beta}}$
    - estimate $\hat{\beta}^{(n+1)}$ by solving $\mathbf{X}^\top\mathbf{W}^{(n)}\mathbf{y} = \mathbf{X}^\top\mathbf{W}^{(n)}\mathbf{X}\hat{\beta}^{(n+1)}$
- scale of estimator $\sigma^{(n)}$

Solving regression with M-Estimator

- scale of estimator $\sigma^{(n)}$ for the update: $w_i^{(n)} = \dfrac{\frac{\partial \rho}{\partial u}}{y_i - \mathbf{x}_i^\top \hat{\beta}}$
  - Median Absolute Deviation (MAD): $\sigma^{(n)} = 1.4826\ \text{median}_i \,|\, r_i^{(n)}(x_i, \hat{\beta}^{(n-1)}) \,|$
  - Huber's proposal 2. Solve for $\sigma^{(n)}$: $$\sum_i \chi\left(\frac{r_i^{(n)}(x_i, \hat{\beta}^{(n-1)})}{\sigma^{(n)}}\right) = Nk_2$$
    - $\chi(u) = \min(|u|, k_1)^2$
    - $k_1 > 0$, k_2 good for normal distributiion,

Robust Linear Regression

- M-Estimators
- Minimization Function with M-Estimators
- Solving regression with M-Estimators

Applied Machine Learning

Robust Linear Regression