annotated version

Machine Learning Course - CS-433

Loss Functions

Cost-

Sept 11, 2024

Martin Jaggi
Last updated on: September 10, 2024

credits to Mohammad Emtiyaz Khan

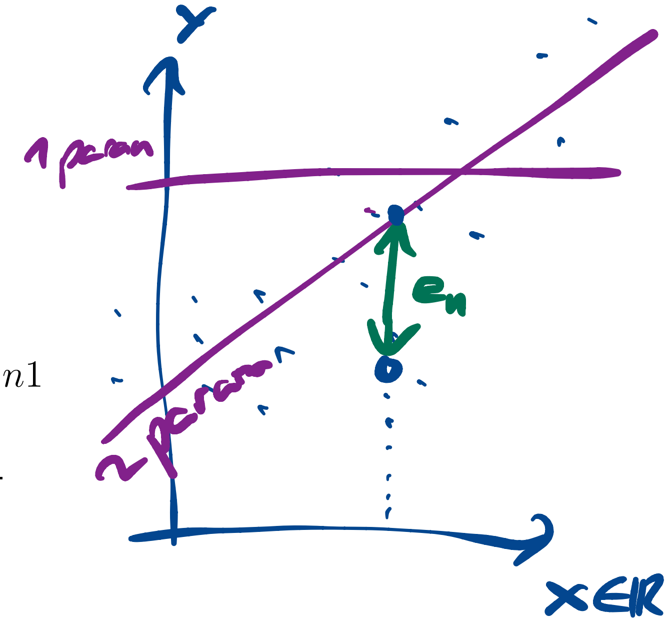
EPFL

Motivation

Consider the following models.

1-parameter model: $y_n \approx w_0$
2-parameter model: $y_n \approx w_0 + w_1 x_{n1}$

How can we estimate (or guess) values of $\mathbf{w}$ given the data $\mathcal{D}$?



What is a loss function?

A loss function (or energy, cost, training objective) is used to learn parameters that explain the data well. The loss function quantifies how well our model does - or in other words how costly our mistakes are.

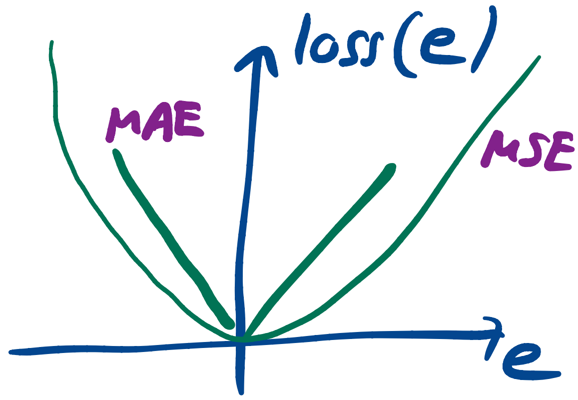
error

$$e_n = y_n - f_{\mathbf{w}}(x_n)$$

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} \text{loss}(e_n)$$

Two desirable properties of loss functions

When the target y is real-valued, it is often desirable that the cost is symmetric around 0, since both positive and negative errors should be penalized equally.



Also, our cost function should penalize "large" mistakes and "very-large" mistakes similarly.

Statistical vs computational trade-off

If we want better statistical properties, then we have to give-up good computational properties.

Mean Square Error (MSE)

MSE is one of the most popular loss functions.

$$\text{MSE}(\mathbf{w}) := \frac{1}{N} \sum_{n=1}^{N} \Big[\overbrace{y_n - f_{\mathbf{w}}(\mathbf{x}_n)}^{e_n}\Big]^2$$

Does this loss function have both mentioned properties?

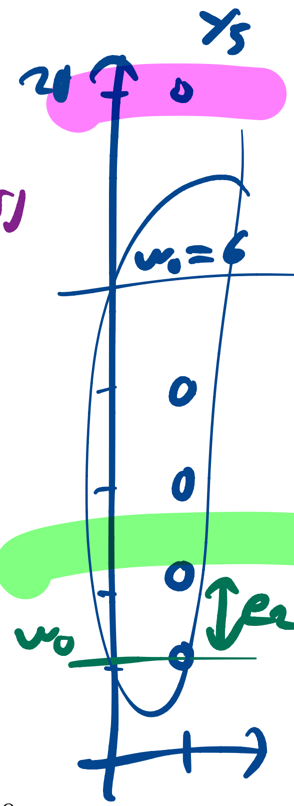
An exercise for MSE

Compute MSE for 1-param model:

$$\mathcal{L}(w_0) := \frac{1}{N} \sum_{n=1}^{N} [\underbrace{y_n - w_0}_{e_n}]^2$$

$w_0 =$	1	2	3	4	5	6	7
$y_1 = 1$	0^2	1^2	2^2	3^2	4^2	5^2	
$y_2 = 2$	1^2	0^2	1^2				
$y_3 = 3$	2^2						
$y_4 = 4$	3^2						
$\text{MSE}(\mathbf{w}) \cdot N$	14	6	6	14	30	54	
$y_5 = 20$	19^2	18^2	17^2	16^2			
$\text{MSE}(\mathbf{w}) \cdot N$					255	250	255

loss



Some help: $19^2 = 361, 18^2 = 324, 17^2 = 289, 16^2 = 256, 15^2 = 225, 14^2 = 196, 13^2 = 169$.

Outliers

Outliers are data examples that are far away from most of the other examples. Unfortunately, they occur more often in reality than you would want them to!

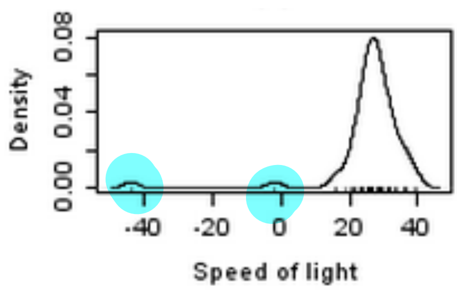
MSE is not a good loss function when outliers are present.

Here is a real example on speed of light measurements

(Gelman's book on Bayesian data analysis)

28	26	33	24	34	-44	27	16	40	-2
29	22	24	21	25	30	23	29	31	19
24	20	36	32	36	28	25	21	28	29
37	25	28	26	30	32	36	26	30	22
36	23	27	27	28	27	31	27	26	33
26	32	32	24	39	28	24	25	32	25
29	27	28	29	16	23				

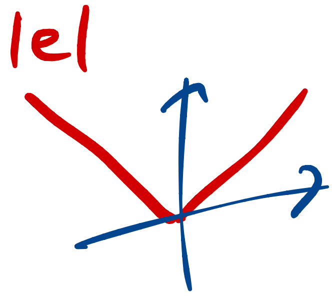
(a) Original speed of light data done by Simon Newcomb.



(b) Histogram showing outliers.

Handling outliers well is a desired *statistical* property.

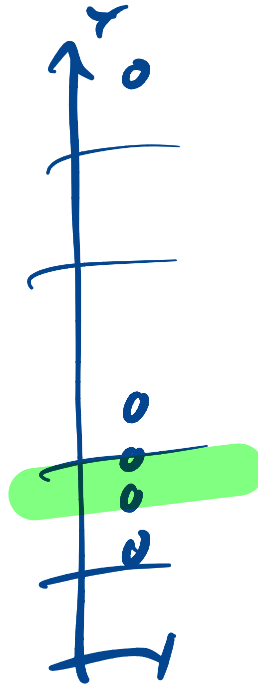
Mean Absolute Error (MAE)



$$\text{MAE}(\mathbf{w}) := \frac{1}{N}\sum_{n=1}^{N}\Big|\overbrace{y_n - f_{\mathbf{w}}(\mathbf{x}_n)}^{e_n}\Big|$$

Repeat the exercise with MAE.

	1	2	3	4	5	6	7
$y_1 = 1$	0	1	2	3	4	5	
$y_2 = 2$	1	0	1	2			
$y_3 = 3$	2	1	0	1			
$y_4 = 4$	3	2	1	0			
$\text{MAE}(\mathbf{w}) \cdot N$	6	4	4	6	10	14	
$y_5 = 20$	19	18	17				
$\text{MAE}(\mathbf{w}) \cdot N$	25	22	21	22	25		

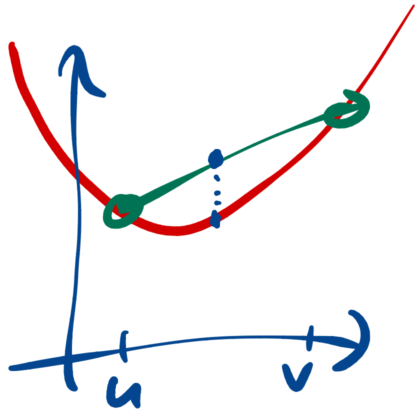


Can you draw MSE and MAE for the above example?

good for computational efficiency

Convexity

Roughly, a function is convex iff a line segment between two points on the function's graph always lies above the function.



A function $h(\mathbf{u})$ with $\mathbf{u} \in \mathbb{R}^D$ is convex, if for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ and for any $0 \leq \lambda \leq 1$, we have:

$$h(\lambda \mathbf{u} + (1-\lambda)\mathbf{v}) \leq \lambda h(\mathbf{u}) + (1-\lambda) h(\mathbf{v})$$

A function is strictly convex if the inequality is strict.

Importance of convexity

A strictly convex function has a unique global minimum $\mathbf{w}^\star$. For convex functions, every local minimum is a global minimum.

Sums of convex functions are also convex. Therefore, MSE combined with a linear model is convex in $\mathbf{w}$.

$\text{loss}(y_n - f_{\mathbf{w}}(\mathbf{x}))$

$x^T \mathbf{w}$

Convexity is a desired *computational* property.

$$\mathcal{L}(w) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_n(w)$$

$$\mathcal{L}_n(w) = |\underbrace{y_n - x^\top w}_{e_n}|$$

Can you prove that fitting a linear model with MAE loss is convex? (as a function of the parameters $\mathbf{w} \in \mathbb{R}^D$, for linear regression $f_{\mathbf{w}}(\mathbf{x}) := f(\mathbf{x}, \mathbf{w}) := \mathbf{x}^\top \mathbf{w}$)

Computational VS statistical trade-off

So which loss function is the best?

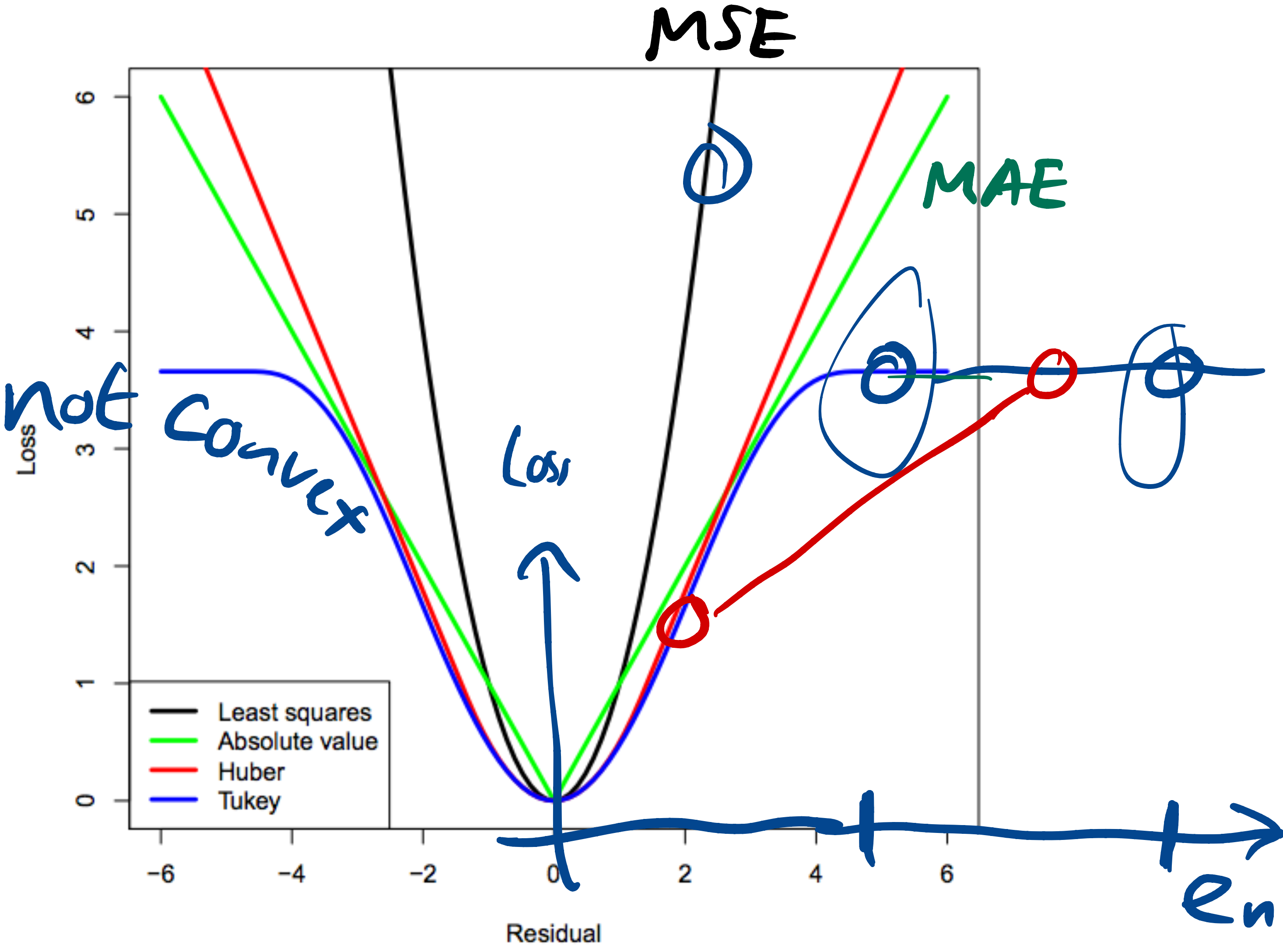


Figure taken from Patrick Breheny's slides.

If we want better statistical properties, then we have to give-up good computational properties.

Additional Reading

Other loss functions

Huber loss

$$Huber(e) := \begin{cases} \frac{1}{2}e^2 & \text{, if } |e| \leq \delta \\ \delta|e| - \frac{1}{2}\delta^2 & \text{, if } |e| > \delta \end{cases} \tag{1}$$

Huber loss is convex, differentiable, and also robust to outliers. However, setting δ is not an easy task.

Tukey's bisquare loss (defined in terms of the gradient)

$$\frac{\partial \mathcal{L}}{\partial e} := \begin{cases} e\{1 - e^2/\delta^2\}^2 & \text{, if } |e| \leq \delta \\ 0 & \text{, if } |e| > \delta \end{cases} \tag{2}$$

Tukey's loss is non-convex, but robust to outliers.

Additional reading on outliers

- Wikipedia page on "Robust statistics".
- Repeat the exercise with MAE.
- Sec 2.4 of Kevin Murphy's book for an example of robust modeling

Nasty loss functions: Visualization

See Andrej Karpathy's Tumblr post for many loss functions gone "wrong" for neural networks. http://lossfunctions.tumblr.com/.