* Empirical Risk Minimization

A lot of ML algorithms involve Minimization of a function over parameters w such that it contains two components : loss function and regularization

$$\min_{w} \left[\underbrace{\frac{1}{n} \sum_{i=1}^{n} l(h_w(x_i), y_i)}_{\text{loss function}} + \underbrace{\lambda r(w)}_{\text{Regularizer}} \right]$$

loss function is a continuous function which penalizes training error, while regularizer penalizes model complexity. l is a function of parameters & data points, while regularizer is function of only parameters

Empirical risk minimization is the idea where we want to decrease loss on training data, but keep model simple.

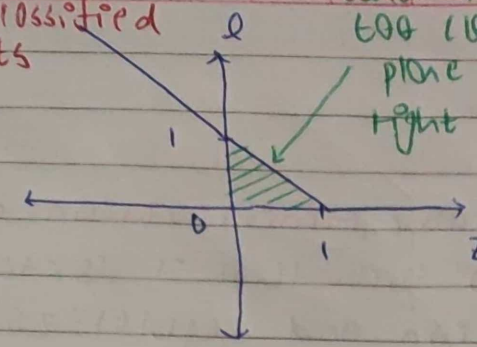
[I] Binary classification loss function

(1.1) Hinge loss $\max\left[1 - h_w(x_i) y_i, 0\right]^p$

This loss is used in SVMs, if $p=1$; standard SVM, if $p=2$; squared hingeless SVM and becomes differentiable

The standard SVM represents the size of error in loses when point is very near to hyperplane or on wrong side

linear penalty for misclassified points

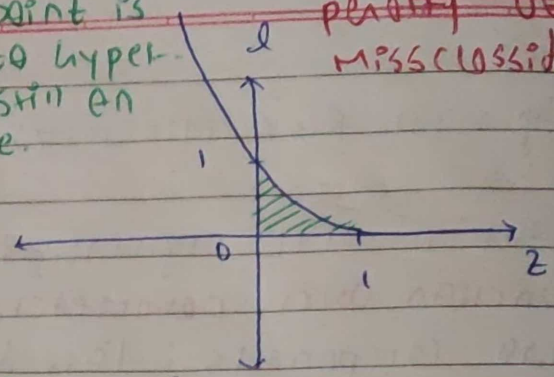


This region represents when the point is too close to hyper plane but still on right side.

P = 1

$\max(1-z, 0)$

squared penalty for misclassified points



P = 2

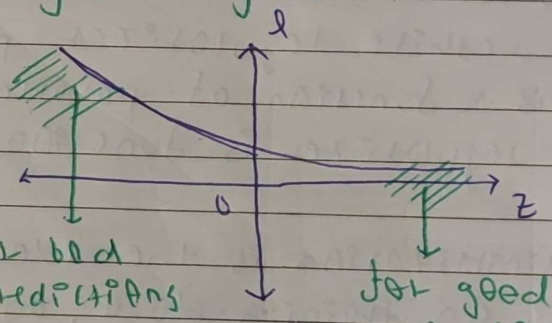
$[\max(1-z, 0)]^2$

(1.2) logistic loss

$\log(1 + e^{-y_i h_w(x_i)})$

used in logistic regression

This is more straight line than curvy

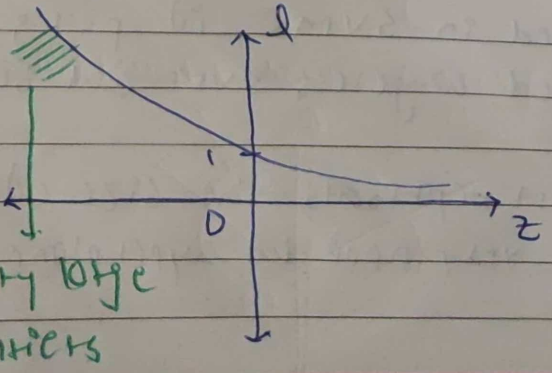


$\log(1 + e^{-z})$

(1.3) Exponential loss

$e^{-y_i h_w(x_i)}$

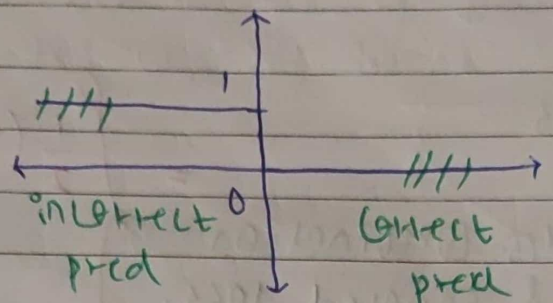
This is very aggressive loss function and it penalizes outliers heavily. Used in adaboost.



e^{-z}

(1.4) 0/1 loss $\delta[\text{sign}(h_\omega(x_i)) \neq y_i]$

This is used for actual classification loss.



$\delta[\text{sign}(z), y_i]$

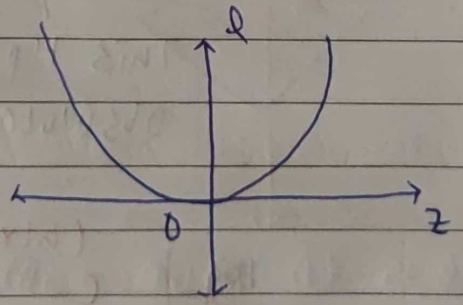
OR

$\delta[y_i \text{ sign}(z) \neq +1]$

[II] Regression loss functions

(2.1) Squared loss (OLS) $(h(x_i) - y_i)^2$

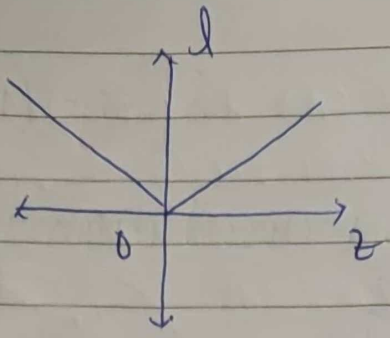
Squared loss is a measure of average/mean. It is differentiable & easy to optimize, but is sensitive to outliers. This is NOT ideal for house-price prediction or average placement etc since outliers dominate.



z^2

(2.2) Absolute loss $|h(x_i) - y_i|$

This is a measure of median. It is NOT differentiable at 0, but is less sensitive to outliers making it good for house price, median salary.

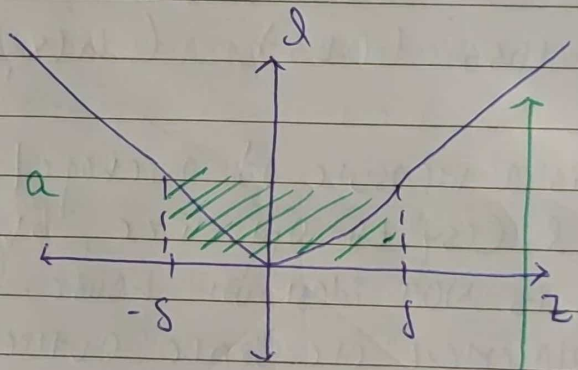


$|z|$

(2.3) Huber loss

$$= \begin{cases} \frac{1}{2}(h(x_i) - y_i)^2 & \text{if } |h(x_i) - y_i| < \delta \\ \delta\left(|h(x_i) - y_i| - \frac{\delta}{2}\right) & \text{otherwise} \end{cases}$$

Huber loss provides combination of absolute loss & squared loss. This is differentiable, resistant to outliers and has optimization properties.



This region is a parabola like ols

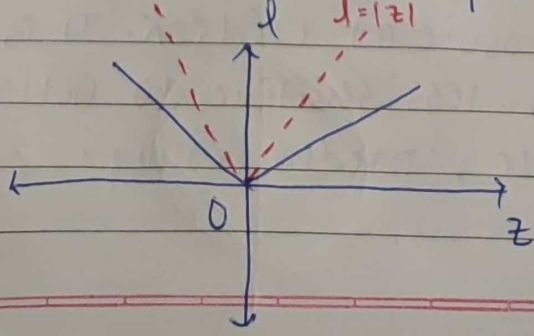
$$= \begin{cases} \frac{1}{2}z^2 & \text{for } |z| \le \delta \\ \delta\left(|z| - \frac{\delta}{2}\right) & \end{cases}$$

This represents straight line of absolute error

(2.4) log cosh loss

$$\log\left[\frac{e^{(h(x_i) - y_i)} + e^{-(h(x_i) - y_i)}}{2}\right]$$

This is twice differentiable and very similar to Huber loss.



$$\log\left[\frac{e^z + e^{-z}}{2}\right]$$

[III] Regularizers

what we try to do is :

regularization term

$$\min_{w,b} \left[\sum_{i=1}^{n} d(h_w(x_i), y_i) + \lambda r(w) \right]$$

This can be rephrased using lagrangian as :-

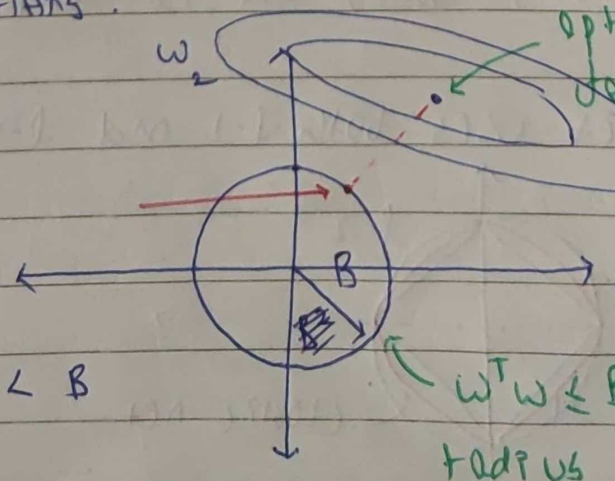
$$\min_{w,b} \left[\sum_{i=1}^{n} d(h_w(x_i), y_i) \right] \quad \text{s.t.} \quad r(w) \leq B$$

Thus, for each λ, there exist a B, where we can show $r(w) \leq B$.

Thus, what this means geometrically is that on one hand, we want the minimum of loss function & we want to restrict w such that $r(w) \leq B$.

(3.1) L2 Regularization $\quad w^T w$ OR $\|w\|_2^2$

This is differentiable & convex in nature. However, this uses weights on all features, i.e. it provides dense solutions.



Our w

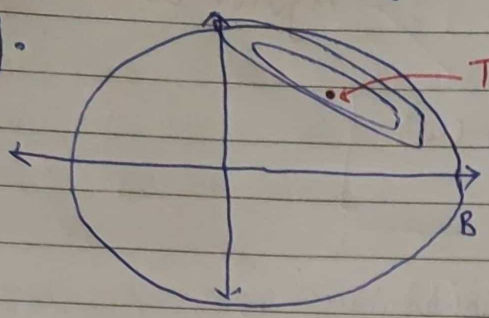
what we want is w in that restricted domain only ; even though optimal is outside

$w_1^2 + w_2^2 < B$

$w^T w \leq B$ leads to a circle of radius

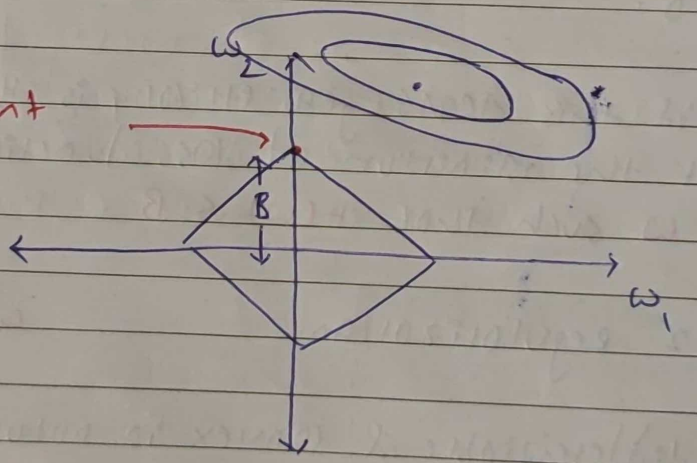
if we increase our B too much, i.e. decrease our λ too much, regularization will have no effect.

eg.



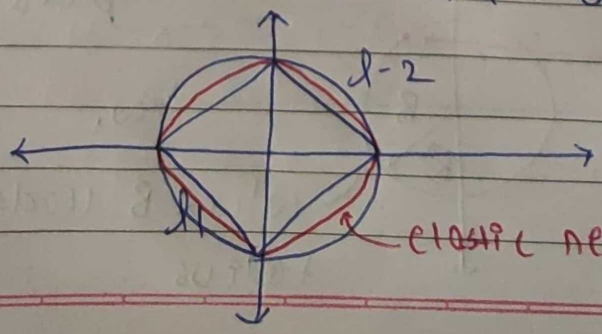
(3.2) l_1 regularization. $\|w\|_1$

This is not strictly convex or differentiable at 0. But, due to its shape, it allows us to obtain sparse solutions.



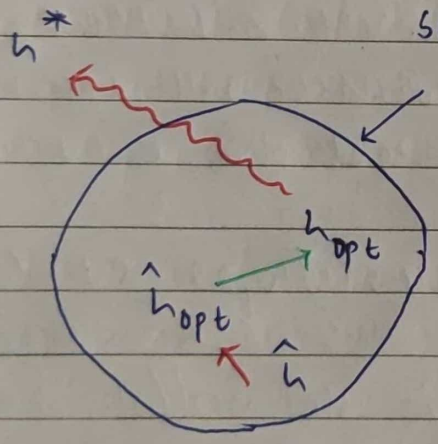
This methods allows to prune some weights, thus removing the features which are not very useful.

Note: Elastic Net uses both l_1 and l_2 regularization.



Since l_1 deals with absolute values, it can afford putting all weights on one dimension and keeping others 0. This is because [2 0 0 0] and [0.5 0.5 0.5 0.5] leads to some sum. But l_2 can't do the same. Here; [2 0 0 0] leads to 4, but [0.5 0.5 0.5 0.5] leads to 1. Hence it has to spread out and is more robust to failures.

- let h^* ⟶ Optimal predictor
  h_{opt} ⟶ optimal hypothesis that can be produced by Algorithm A.
  $\hat{h}_{opt}$ ⟶ optimal hypothesis that can be produced by Algorithm A on Data D.
  $\hat{h}$ ⟶ hypothesis found by Algorithm A



search space H.

$$\varepsilon(\hat{h}) - \varepsilon(h^*)$$

$$= \boxed{\varepsilon(h_{opt}) - \varepsilon(h^*)} \quad \text{Approximation}$$

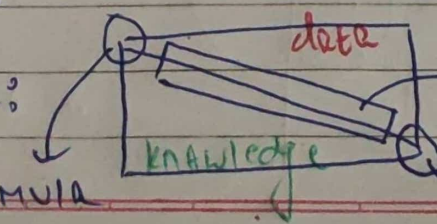
$$+ \boxed{\varepsilon(\hat{h}_{opt}) - \varepsilon(h_{opt})} \quad \text{Estimation}$$

$$+ \boxed{\varepsilon(\hat{h}) - \varepsilon(\hat{h}_{opt})} \quad \text{Optimization}$$

A is responsible for optimization error.

These are the 3 things where we provide knowledge. For eg. Bias.

Note:



ML = theda data + theda knowledge

eg. DL is best for ↑data & ↓knowledge.