Fondamentaux de l'Apprentissage Automatique

Lecturer: Yann Chevaleyre
Scribe: Lyna Bouikni

Lecture n°2 #
05/10/2023

1 Introduction

The overall goal of this class is to understand that solving a classification problem inherently aims at minimizing the $0/1$ loss. However, given its non-convex nature and associated complexities, we pivot to alternative loss functions including cross entropy and other surrogate losses. The selection criteria and rationale behind choosing one loss function over another will be the central focus of this lecture.

2 Linear classification with the $0/1$ loss

Given a binary classification problem, the aim is to distinguish data points into one of two classes. Let's define our data set as :

$$S = \{(x_1, y_1), \ldots, (x_N, y_N)\} \tag{1}$$

where $x \in \mathbb{R}^d$ and $y = \hat{y} \in \{0, 1\}$.

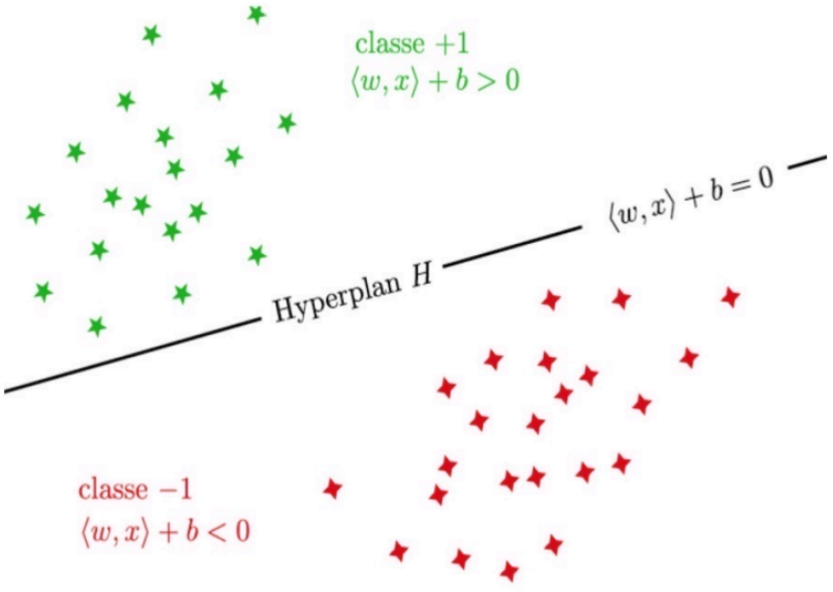


FIGURE 1 – Binary classification setting

The goal is to classify each data point as either green (1) or red (0). To achieve this, we'll leverage the linear function class $\mathcal{F}$, defined by :

$$\mathcal{F} = \left\{ x \mapsto \mathbb{1}\left[\theta^\top x + b \geqslant 0\right] : \theta \in \mathbb{R}^d, b \in \mathbb{R} \right\}. \tag{2}$$

The set $\mathcal{F}$ represents the class of all linear classifiers, and we aim to find the function f from the class $\mathcal{F}$ of linear classifiers that best fits our data set S .

To formalize our approach, consider the 0/1 loss, defined as :

$$\ell^{0,1}(\hat{y}, y) = \mathbb{1}[\hat{y} \neq y] \tag{3}$$

This loss function penalizes by 1 for misclassification and 0 otherwise.

Our primary objective is the minimization of these misclassifications. Grounded in the Empirical Risk Minimization (ERM) principle, this is formally expressed as :

$$\underset{f \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{N} \underbrace{\ell^{0/1}\left(f(x_i), y_i\right)}_{\mathbb{1}[f(x_i) \neq y_i]} \tag{4}$$

where $\mathbb{I}[.]$ is the indicator function, returning 1 when the prediction $f(x_i)$ does not match the true label y_i, and 0 otherwise.

This is equivalent to saying, we aim to identify the optimal hyperplane parameterized by θ and b that minimizes the empirical risk, quantified by the total number of misclassifications across both classes. This is formally expressed as :

$$\underset{\theta, b}{\arg\min} \left(\sum_{\{i | y_i = 1\}} \mathbb{1}\left[\theta^\top x_i + b < 0\right] + \sum_{\{i | y_i = 0\}} \mathbb{1}\left[\theta^\top x_i + b \geq 0\right] \right) \tag{5}$$

The objective function above encapsulates two summations : the first accounts for errors when true labels are $y_i = 1$ and the classifier mispredicts them on the wrong side of the hyperplane, while the second sum captures analogous errors for the $y_i = 0$ class. Thus, our goal is to optimize θ and b such that this cumulative error is minimized.

The challenge lies in determining the optimal parameters θ and b that minimize errors for both classes. When the two classes are linearly separable, as depicted in Fig. 1, the task becomes a straightforward linear programming problem. However, complications arise when the classes aren't neatly separable (Refer to fig. 2). In such cases, the problem becomes NP-hard.

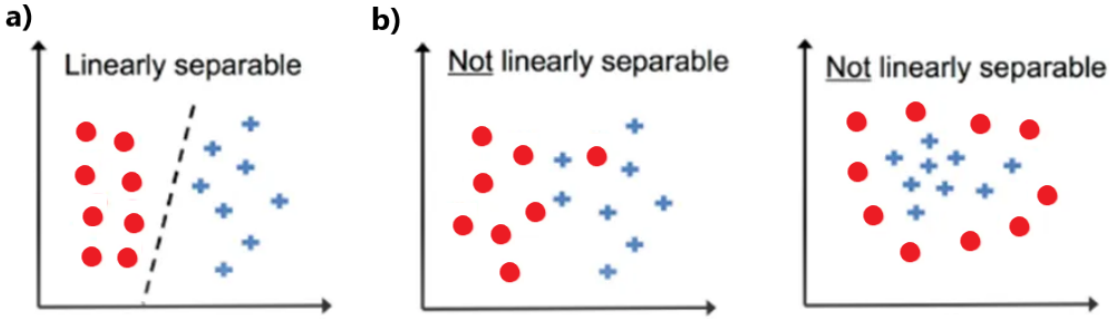


FIGURE 2 – Visualization of classes : Linearly Separable vs. Non-linearly Separable

To illustrate the complexity : even when we know there exists a classifier that achieves an error rate of less than ϵ, finding a classifier that guarantees an error rate just slightly better than random guessing (e.g., better than 50% error rate) can be NP-hard. Thus, even if a near-perfect classifier is known to exist, the quest for even moderately good classifiers can be computationally challenging.

3 Linear classification with the logistic regression framework :

The inherent challenge in our prior approach to binary classification is its combinatorial nature ; predictions strictly fall into either 0 or 1. This discrete decision-making structure can result in NP-hard problems, as previously discussed.

To overcome this limitation, we transition from a strict binary classification mechanism to a continuous function, paving the way for more nuanced decision boundaries. Instead of a rigid classifier where one side of a hyperplane corresponds to class 0 and the other to class 1, we introduce a continuous prediction model.

The idea is to replace the binary classifier with a continuous function whose parameters we can optimize as a convex optimization problem such as the sigmoid function (see fig. 3).

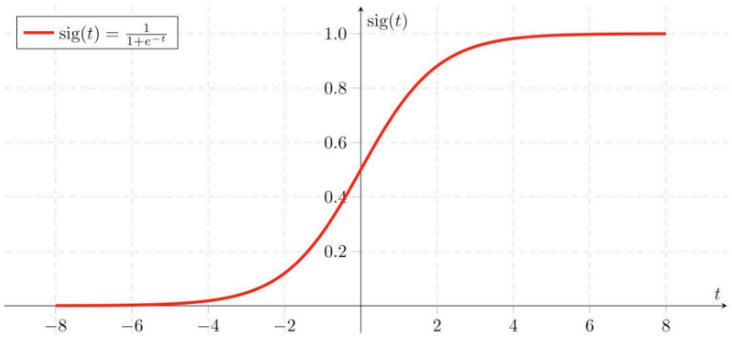


FIGURE 3 – Sigmoid function

Example :

Consider a one-dimensional dataset representing male and female individuals based on their height. Employing the 0/1 loss function would mean determining an absolute height threshold where heights above are deemed 'male' and below as 'female'. This strict delineation is not always optimal. A smoother function, such as the sigmoid, provides probabilistic predictions instead of fixed classifications. For a specific height, the prediction might indicate a higher probability of the individual being male (red point) as opposed to female (blue point) (see fig. 4).

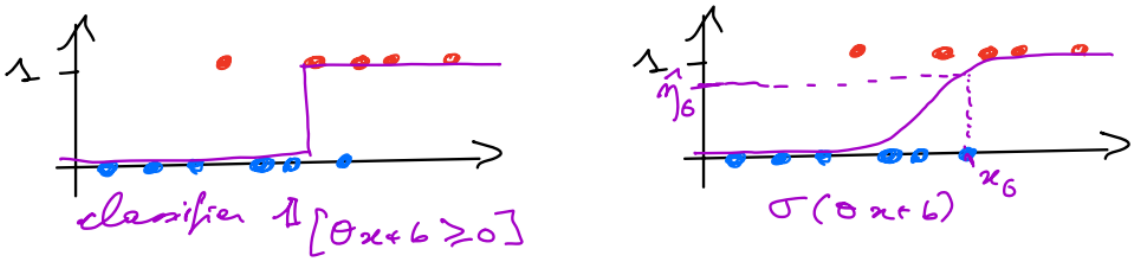


FIGURE 4 – Comparison of Indicator and Sigmoid functions as classifiers

3.1 Setting the Notation :

As we proceed in this course, each data point will be assigned a *score* for classification. Given our hyperplane characterized by parameters θ and b :

1. The score for each data point x_i is :

$$\hat{y}_i = \theta^T x_i + b$$

2. Utilizing this score, we determine the associated probability of the point belonging to a specific class, termed the Class Probability Estimate (CPE) :

$$\hat{\eta}_i = \sigma(\hat{y}_i) = \frac{1}{1 + e^{-\hat{y}_i}}$$

Where :

- $\hat{\eta}_i$ is interpreted as the conditional probability $P(y = 1 \mid X = x_i, \theta, b)$, representing the likelihood that the observed data point x_i belongs to class 1 given the model parameters.

- $1 - \hat{\eta}_i$ is its complementary probability, interpreted as $P(y = 0 \mid X = x_i, \theta, b)$, representing the likelihood of the observed point x_i belonging to class 0 given the model parameters.

3. The predicted class is defined as :

$$\hat{c}_i = \mathbf{1}_{[\hat{y}_i \geq 0]} = \mathbf{1}_{\left[\hat{\eta}_i \geq \frac{1}{2}\right]}$$

3.2 Objective : Determining optimal parameters θ and b

Initial Approach :

A straightforward approach would be to classify each point based on whether its score surpasses a threshold. This aligns with our prior challenge :

$$\hat{\theta}, \hat{b} = \text{argmin} \sum_{i=1}^{N} \mathbf{1}_{[\hat{C}_i \neq y_i]}$$

(Note : This method replicates our earlier problem, rendering it computationally complex.)

Alternative Approach :

A more refined, yet still sub-optimal strategy is :

$$\hat{\theta}, \hat{b} = \text{argmin} \sum_{i=1}^{N} (\hat{\eta}_i - y_i)^2$$

(Note : Although this objective function is continuous, it's non-convex, which can complicate optimization.)

Exercise 1 : Non-Convexity of the Objective Function

Given the objective function :

$$\text{argmin} \sum_{i=1}^{N} (\hat{\eta}_i - y_i)^2$$

demonstrate, using a single data example $(0, 0)$, that this function is not convex.

Solution :

Recall the expressions :

$$\hat{y}_i = \theta^T x_i + b$$
$$\hat{\eta}_i = \sigma(\hat{y}_i) = \frac{1}{1 + e^{-\hat{y}_i}}$$

The given function can be written as :

$$L(\theta, b) = \sum_{i=1}^{N} (\hat{\eta}_i - y_i)^2$$

With the condition $x_1 = y_1 = 0$, thus :

$$\hat{y_1} = b$$
$$\hat{\eta_1} = \sigma(b) = \frac{1}{1 + e^{-b}}$$

Leading to the minimization problem :

$$\text{argmin}_b \, \sigma(b)^2$$

To confirm the objective's non-convexity, we need to examine the behavior of $L(\theta, b) = \left(\frac{1}{1+e^{-b}}\right)^2$ with variations in θ and b.

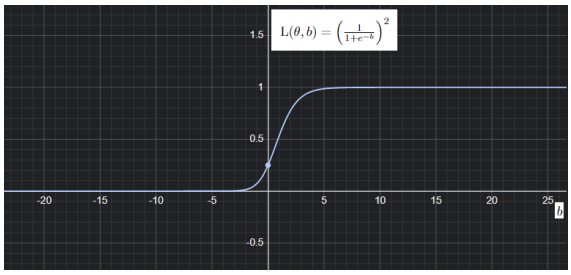


FIGURE 5 – The behavior of $L(\theta, b)$ as b varies

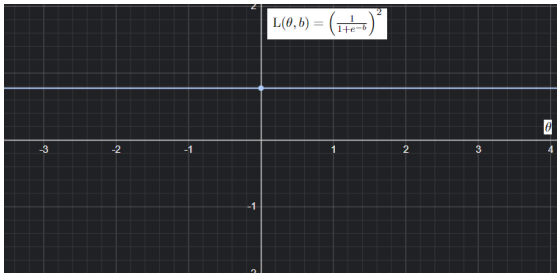


FIGURE 6 – The behavior of $L(\theta, b)$ as θ varies

Hence, the objective function is not convex.

4 Linear Classification via Logistic Regression : Choice of Loss Function

The probabilistic interpretation of logistic regression provides insight into the selection of an appropriate loss function. The likelihood for model parameters θ and b is given by :

$$\mathcal{L}(\theta, b) = \prod_{i=1}^{N} p\left(y = y_i \mid X = x_i, \theta, b\right) = \prod_{i:y_i=1} \hat{\eta}_i \times \prod_{i:y_i=0} \left(1 - \hat{\eta}_i\right). \tag{6}$$

Maximizing this likelihood with respect to θ and b is equivalent to minimizing the negative log-likelihood $\mathcal{NL}(\theta, b)$:

$$\mathcal{NL}(\theta, b) = -\log \mathcal{L}(\theta, b) \tag{7}$$
$$= -\sum_{i:y_i=1} \ln \hat{\eta}_i - \sum_{i:y_i=0} \ln (1 - \hat{\eta}_i) \tag{8}$$
$$= \sum_{i=1}^{N} [-y_i \ln \hat{\eta}_i - (1 - y_i) \ln (1 - \hat{\eta}_i)] \tag{9}$$
$$= l^{CE} (\hat{\eta}_i, y_i) \text{ , where } l^{CE} \text{ is the cross-entropy loss.} \tag{10}$$

Thus, the logistic regression optimization problem can be succinctly represented as :

$$(\hat{\theta}, \hat{b}) = \text{argmin}_{\theta,b} \sum_{i=1}^{N} l^{CE} (\hat{\eta}_i, y_i) \,. \tag{11}$$

Exercise 2 :

— Draw $l^{CE}(\hat{\eta}, 1)$ and $l^{CE}(\hat{\eta}, 0)$
— Based on the drawing, show $\text{argmin}_{\hat{\eta}}\, l^{CE}(\hat{\eta}, 0)$ and $\text{argmin}_{\hat{\eta}}\, l^{CE}(\hat{\eta}, 1)$.

Solution :

$l^{CE}(\hat{\eta}, 0) = -\ln(1 - \hat{\eta})$
$l^{CE}(\hat{\eta}, 1) = -\ln \hat{\eta}$

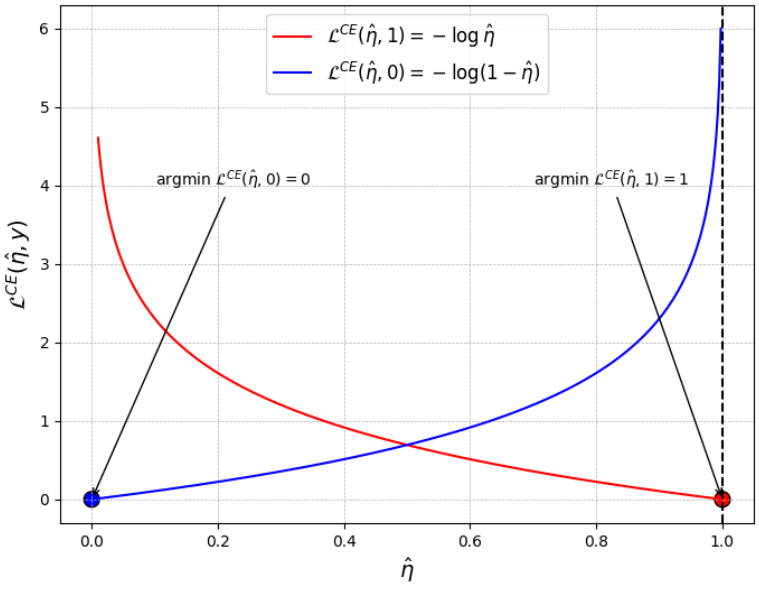


FIGURE 7 – Cross-Entropy loss functions for Binary classification

From the graphs, we deduce :

$\text{argmin}_{\hat{\eta}}\, l^{CE}(\hat{\eta}, 0) = 0$
$\text{argmin}_{\hat{\eta}}\, l^{CE}(\hat{\eta}, 1) = 1$

5 Multiclass Logistic Regression

Multiclass logistic regression generalizes the binary logistic regression to accommodate multiple classes. A prevalent and simple approach involves using hyperplanes to segregate each class.

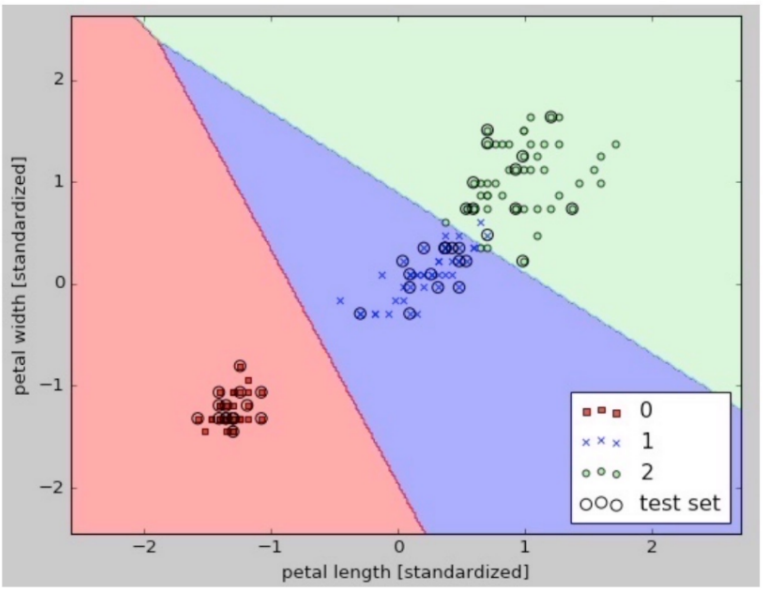


FIGURE 8 – Visualization of hyperplanes for multiple classes.

5.1 Notation

Given K classes, the set of classes is represented as : $y = \{1, \ldots, K\}$.

- For each $k \in y$, we define weights $\theta_k \in \mathbb{R}^d$ and bias $b_k \in \mathbb{R}$.

- The score vector of x_i is $(\hat{y}_{i,1}, \ldots, \hat{y}_{i,K})$, computed as $\left(\theta_1^T x_i + b_1, \ldots, \theta_K^T x_i + b_K\right)$.

5.2 Softmax Function

In the multiclass setting, the sigmoid function is replaced with the softmax to map score vectors to probability distributions :

$$\text{softmax}(\mathbf{t}) = \frac{1}{\sum_{k=1}^{K} e^{t_k}} \left(e^{t_1}, \ldots, e^{t_K}\right) \tag{12}$$

Then,

- The Class Probability Estimate (CPE) for x_i is : $\hat{\eta}_i = \text{softmax}\left(\hat{y}_{i,1}, \ldots, \hat{y}_{i,K}\right)$.
- The predicted class is given by : $\hat{c}_i = \text{argmax}_{k \in \{1,\ldots,K\}}\, \hat{y}_{i,k}$.
- The multiclass cross-entropy loss is defined as : $l^{CE}\left(\hat{\eta}_i, y_i\right) = \sum_{k=1}^{K} -\mathbf{1}_{[y_i=k]} \ln \hat{\eta}_{i,k}$.

Example for the multiclass setting :

Consider three classes, represented as $y = \{\text{dog}, \text{cat}, \text{mouse}\}$, and a data point $X = (2, 3)$. We determine the scores for this point using three hyperplanes corresponding to each class.

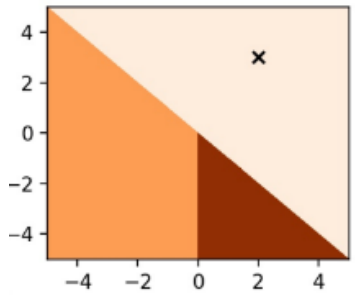


FIGURE 9 – Hyperplanes for multi-class classification

The following table summarizes the parameters, scores, class probability estimates (CPEs), and the predicted class :

Parameters		
$\boldsymbol{\theta}_{\text{dog}} = (\lambda, \lambda)^T$ $b_{\text{dog}} = 0$	$\boldsymbol{\theta}_{\text{cat}} = (-1, 0)^T$ $b_{\text{cat}} = 0$	$\boldsymbol{\theta}_{\text{mouse}} = (\lambda, -\lambda)^T$ $b_{\text{mouse}} = 0$
Scores		
$\hat{y}_{\text{dog}} = 5$	$\hat{y}_{\text{cat}} = -2$	$\hat{y}_{\text{mouse}} = -1$
CPEs		
$\exp(\hat{y}_{\text{dog}}) = 1.48$ $\sum_c \exp(\hat{y}_c) = 1.49$ $\hat{\eta}_{\text{dog}} = 99.6\%$	$\exp(\hat{y}_{\text{cat}}) = 0.1$ $\hat{\eta}_{\text{cat}} = 0.1\%$	$\exp(\hat{y}_{\text{mouse}}) = 0.4$ $\hat{\eta}_{\text{mouse}} = 0.2\%$
Predicted class		
$\hat{c} = \text{dog}$		

TABLE 1 – Summary of Parameters, Scores, CPEs, and Predicted Class for the Multiclass Setting

6 Two Perspectives on Logistic Regression in Binary Classification

Logistic regression in a binary classification setting can be approached from two distinct perspectives. The first perspective, which we have primarily discussed, focuses on class probability estimates and minimizes the cross-entropy loss. The second perspective emphasizes scores and employs the logistic loss.

View 1 : Cross-Entropy Loss with Class Probability Estimates

In this view, the model's output is treated as the class probability estimate. The appropriate loss function in this scenario is the cross-entropy loss :

$$\hat{\theta}, \hat{b} = \underset{\theta,b}{\operatorname{argmin}} \sum_{i=1}^{N} l^{CE}(\hat{\eta}_i, y_i) \tag{13}$$

where $l^{CE}(\hat{\eta}, y)$ is the cross-entropy loss and $y \in \{0, 1\}$.

View 2 : Logistic Loss with Scores

Alternatively, logistic regression can be perceived as producing scores, denoted by $\hat{y}$. In this view, the logistic loss is used :

$$\hat{\theta}, \hat{b} = \underset{\theta,b}{\operatorname{argmin}} \sum_{i=1}^{N} l^{\text{logistic}}(\hat{y}_i, y_i) \tag{14}$$

$$l^{\text{logistic}}(\hat{y}_i, y_i) = \ln(1 + e^{-y_i \hat{y}_i}) \tag{15}$$

where the loss function $l^{\text{logistic}}(\hat{y}, y)$ is defined for $y \in \{-1, 1\}$.

Remark : Both perspectives are fundamentally equivalent and result in convex optimization problems in terms of θ and b.

<u>Exercise 3</u> : Develop view 1 to show it is identical to view 2

Context **:** View 1 pertains to classes 0 and 1, whereas View 2 is concerned with classes -1 and 1. For the purpose of this exercise, consider a dataset wherein all instances of class 0 are replaced with class -1. Our goal is to express the losses from View 1 using their corresponding expressions from View 2, thereby establishing their equivalence.

Note that the sigmoid function plays a crucial role in this derivation.

Solution :

$$
\begin{aligned}
l^{CE}(\hat{\eta}, 1) &= -\ln \hat{\eta} = -\ln\left(\frac{1}{1+e^{-\hat{y}}}\right) = \ln(1+e^{-\hat{y}}) \\
l^{CE}(\hat{\eta}, 0) &= -\ln(1-\hat{\eta}) = -\ln\left(1-\frac{1}{1+e^{-\hat{y}}}\right) = -\ln\left(\frac{e^{-\hat{y}}}{1+e^{-\hat{y}}}\right) \\
l^{CE}(\hat{\eta}, 0) &= -\ln\left(\frac{1}{1+e^{\hat{y}}}\right) = \ln\left(1+e^{\hat{y}}\right) \\
l^{CE}(\hat{\eta}, y) &= \ln(1+e^{-y\hat{y}}) = l^{\text{logistic}}
\end{aligned}
$$

Where l^{CE} is **the cross entropy loss** for $y \in \{0, 1\}$ and l^{logistic} is **the logistic loss** for $y \in \{-1, 1\}$.

7 Convexity of learning problems

Given the logistic loss function defined as :

$$l(\hat{y}, y) = \ln(1+e^{-y\hat{y}})$$

where $y \in \{-1, 1\}$.

The function is convex in $\hat{y}$.

Proof :

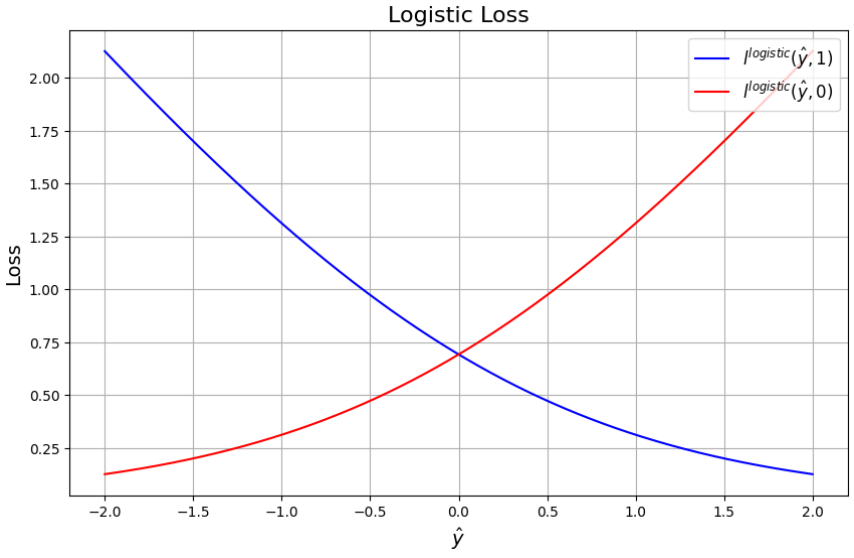


FIGURE 10 – Logistic Loss for y={-1,1}

The overall objective function is defined by :

$$\sum_{i=1}^{N} l(\hat{y}_i, y_i) = \sum_{i=1}^{N} l(\theta^T x_i + b, y_i)$$

which is convex in θ and b.

8 Transitioning from Logistic Regression to Neural Networks : A Dual Perspective

Neural networks, especially those employed for image classification, often culminate in a softmax layer followed by a cross entropy function. These networks can be interpreted or viewed from two distinct perspectives :

1. **Class Probability Estimate (CPE) Perspective :** This perspective focuses on the neural network's ability to produce class probability estimates, denoted as $\hat{\eta}_i$. For instance, the network might estimate the probability of an image being a cat or a dog. Upon generating these estimates, a corresponding CPE loss is applied. This is the essence of the first view.

2. **Score-based Perspective :** In this perspective, the emphasis is on the scores outputted by the neural network, represented as $\hat{y}_i$. Here, a scoring loss function is utilized, such as : the multi-class logistic loss.

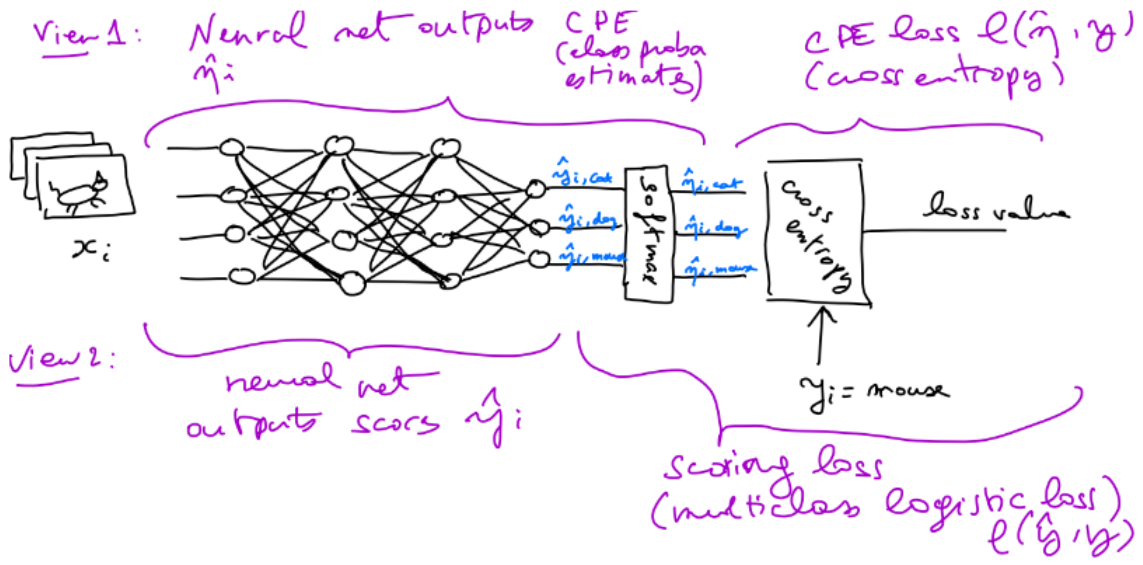


FIGURE 11 – Schematic representation of a neural network elucidating the two aforementioned perspectives.

Intriguingly, these two perspectives are interrelated. The logistic loss is, in essence, a fusion of the softmax function and the cross entropy.

9 Class Probability Estimate (CPE) Losses

The focus thus far has been on datasets ; moving forward, emphasis will be on distributions, a more apt approach for our context. In this discussion, we'll constrain to $y = \{0, 1\}$. Recall, the CPE loss is represented as $l(\hat{\eta}, y)$ (**e.g.**, cross entropy).

9.1 Criteria for Effective CPE Losses

An effective CPE loss should induce a convex learning problem. Additionally, after training, a model should predict with accuracy in areas with known distributions.

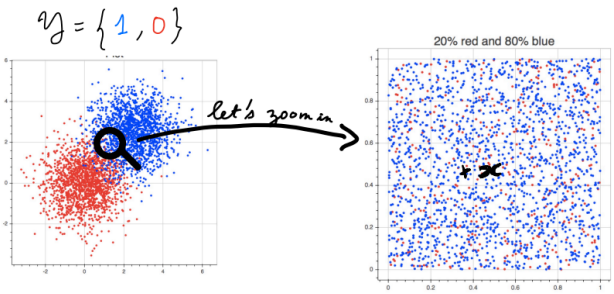


FIGURE 12 – Class distribution visualization with a focused region analysis

Consider a distribution with red and blue points, where blue represents class 1 and red class 0. If a region has 80% blue points, a neural network trained properly should estimate a probability of 80% for class 1 in this region. A well-trained network will optimize $\hat{\eta}$ to yield proper probability estimates.

The average loss in this region can be calculated as :

$$\eta l(\hat{\eta}, 1) + (1 - \eta) l(\hat{\eta}, 0)$$

Examples :

Example 1 : Given a dataset of images indistinguishable by a neural network, comprising 80% mouse and 20% cats, a well-trained network with a good CPE loss should yield :

$$\hat{\eta}_{i,\text{mouse}} = 0.8$$
$$\hat{\eta}_{i,\text{cat}} = 0.2$$

Example 2 : For a data point x in a region where $\eta = P(y = 1|X) = 80\%$, a network trained with an effective CPE loss should predict $\hat{\eta} = 80\%$.

9.2 Risk of CPE-losses

To provide a rigorous understanding of the risk in the context of CPE-losses, we adopt a risk-centric perspective and modify our risk definition to align with the associated loss.

Notation :

- $\hat{\eta} \in [0, 1]$ signifies a probability.

- $\hat{\eta}(.) : \mathbb{R}^d \to [0, 1]$ denotes a CPE prediction function, **e.g,** a neural network with softmax.

Definition 1.

For a given CPE-loss l, the risk of a prediction function $\hat{\eta}(.)$ is formulated as :

$$\mathcal{R}^l(\hat{\eta}(.)) = \mathbb{E}_{x,y\sim p}[l(\hat{\eta}(x), Y)] \tag{16}$$

This formulation is reminiscent of previous discussions, with the key deviation being our transition from classifiers to probability predictors. To attain a comprehensive insight, it's imperative to consider the conditional risk.

9.3 Conditional Risk of CPE losses

To achieve a nuanced understanding of the risk associated with CPE losses, it's imperative to consider the conditional risk, which represents the risk within a specific spatial region.

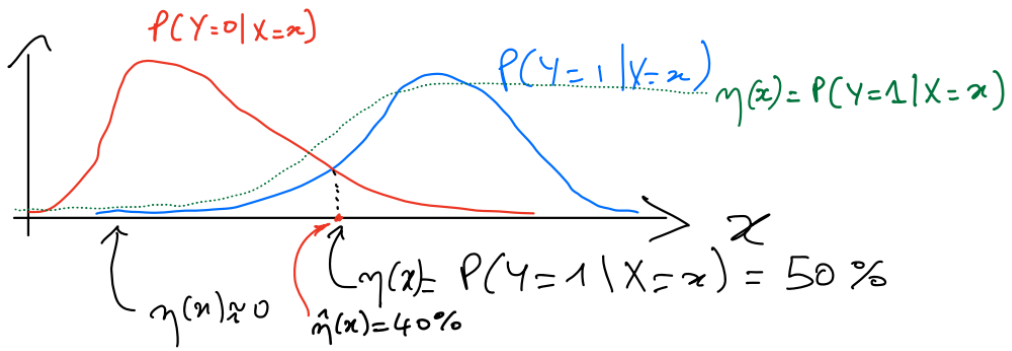


FIGURE 13 – Conditional Risk Visualization

Definition 2. Given $\eta \in [0, 1]$ and $\hat{\eta} \in [0, 1]$, the conditional risk is defined as :

$$C(\hat{\eta}, \eta) = \mathbb{E}_{y\sim \text{Bern}(\eta)}\left[l(\hat{\eta}, y)\right] \tag{17}$$

$$= \eta(x) \times l(\hat{\eta}(x), 1) + (1 - \eta(x)) \times l(\hat{\eta}(x), 0) \tag{18}$$

And the overall risk is :

$$
\begin{aligned}
R(\hat{\eta}(.)) &= \mathbb{E}_{x,y\sim p}\left[l(\hat{\eta}(x), y)\right] && (19)\\
&= \mathbb{E}_x\left[\mathbb{E}_y\left[l(\hat{\eta}(x), y)|X\right]\right] && (20)\\
&= \mathbb{E}_x\left[\eta(x) \times l(\hat{\eta}(x), 1) + (1-\eta(x)) \times l(\hat{\eta}(x), 0)\right] && (21)\\
&= \mathbb{E}_x\left[C(\hat{\eta}(x), \eta(x))\right] && (22)
\end{aligned}
$$

This relationship between conditional risk and overall risk can be further elucidated as :

$$\mathcal{R}^l(\hat{\eta}(x)) = \mathbb{E}_x y(l\hat{\eta}(x), y) = \mathbb{E}_x\mathbb{E}_{Y|x}\left[l(\hat{\eta}(x), y\right] = \mathbb{E}_x\mathcal{C}(\hat{\eta}(x), \eta(x))$$

Where :

$$\eta(x) = P(Y = 1 \mid X = x)$$

Thus, the overall risk is essentially the expected value of the conditional risk across all spatial points, encapsulating both predicted class probabilities and the true class labels.

9.4 Proper CPE losses

A "good" CPE-loss must be a proper loss.

Definition 3. A loss is proper if :

$\forall \eta \in [0,1] \quad \eta \in \operatorname{argmin}_{\hat{\eta}} \mathcal{C}(\hat{\eta}, \eta)$

Exercise 4 :

1. Show that the cross entropy is proper.
2. Show (later) that the mean square error (MSE) is strictly proper.
3. Write the Bayes risk $\left(= \min_f \mathbb{R}^l(f)\right)$ for the case l is proper. Apply it to cross entropy.

Solution :

1. **Showing that the cross entropy is proper :**

   A CPE loss is said to be proper if eta belongs to the argmin of the conditional risk, meaning that :

   We must show that : $\eta = \operatorname{argmin}_{\hat{\eta}} \mathcal{C}(\hat{\eta}, \eta)$

   $$C(\hat{\eta}, \eta) = -\eta \ln \hat{\eta} - (1-\eta)\ln(1-\hat{\eta})$$

Taking the partial derivative with respect to $\hat{\eta}$:

$$\frac{\partial C}{\partial \hat{\eta}} = -\frac{\eta}{\hat{\eta}} + \frac{1-\eta}{1-\hat{\eta}} = 0 \implies \frac{\eta}{\hat{\eta}} = \frac{1-\eta}{1-\hat{\eta}}$$

From the above, we get :

$$\eta(1-\hat{\eta}) = \hat{\eta}(1-\eta) \implies \eta = \hat{\eta} \implies L \text{ is (strictly) proper.}$$

2. **Writing the Bayes risk $\left(= \min_f \mathbb{R}^l(f)\right)$ for the case l is proper and apply it to cross entropy :**

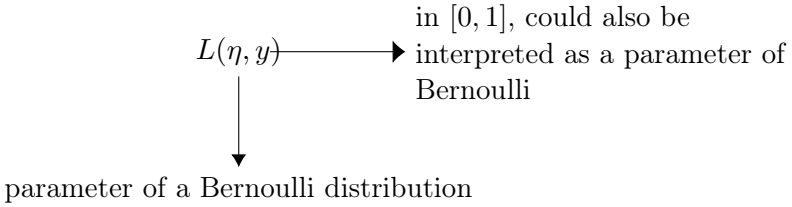
$$\inf_f R'(f) = \mathbb{E}_x \inf_{\hat{\eta}} C(\eta_{\hat{\eta}}, \eta(x)) = \mathbb{E}_x C(\eta(x), \eta(x))$$

For cross entropy :

$$\mathbb{E}_x C(\eta(x), \eta(x)) = \mathbb{E}_x H(\eta(x)) \quad \text{where} \quad H(\cdot) \text{ is entropy.}$$

9.5 More on CPE-losses and the cross-entropy

In the study of neural networks, when addressing the case where both classes represent probabilities, we turn our focus to cross-entropy. Specifically, when defining the cross-entropy function, it encompasses the summation of losses. The output of the neural network, given as this loss, can be conceptualized as a probability, thus acting as a parameter for the Bernoulli distribution. Furthermore, η can also be interpreted as a parameter of the Bernoulli distribution.



Therefore, the loss function $l(\hat{\eta}, y)$, where η is in [0,1], serves as a measure of the dissimilarity between two Bernoulli distributions.

Example : $l(10\%, 0)$ can be interpreted as a dissimilarity between two distributions : Ber(10%) and Ber(0%).

There are plenty of ways to measure the dissimilarity between distributions, the simplest is the total variation distance but it's very inconvenient because it cannot be expressed as

an expectation. A standard way to compare discrete distributions is the Kullback-Leibler divergence, which measures how far apart are two distributions.

For two discrete distributions p and q on $\mathcal{Y}$:

$$KL(p||q) = \sum_{y \in \mathcal{Y}} p(y) \log \frac{p(y)}{q(y)} \tag{23}$$

Properties :

$$\begin{aligned} &KL(p||p) = 0 \\ &KL(p||q) > 0 \quad \text{if } p \neq q \\ &KL(p||q) \neq KL(q||p) \end{aligned}$$

Given the two Bernoulli distributions, $\text{Ber}(\hat{\eta})$ (representing the predicted probabilities) and $\text{Ber}(y)$ (representing the true outcomes), we can compute the Kullback-Leibler (KL) divergence between them to measure the difference (compare them) :

$$\begin{aligned} KL(\text{Ber}(y), \text{Ber}(\hat{\eta})) &= y \ln \left(\frac{y}{\hat{\eta}} \right) + (1-y) \ln \left(\frac{1-y}{1-\hat{\eta}} \right) \\ &= \underbrace{-y \ln \hat{\eta} - (1-y) \ln(1-\hat{\eta})}_{\text{Cross-Entropy, } L^{CE}(\hat{\eta}, y)} + \underbrace{y \ln y + (1-y) \ln(1-y)}_{\text{Negative Entropy of Ber}(y)} \end{aligned}$$

We assume that $0 \ln 0 = 0$ for the purpose of this computation.

Thus, we can simplify the KL divergence to :

$$l^{CE}(\hat{\eta}, y) = KL(\text{Ber}(y), \text{Ber}(\hat{\eta})) \tag{24}$$

This showcases that the cross-entropy loss between our predictions and true outcomes can be viewed as the KL divergence between the two respective Bernoulli distributions.

10 Scoring Losses

Notation :

- $Y = \{-1, 1\}$: Binary class labels.
- $l(\hat{y}, y)$: Logistic loss function
- $\hat{y}_i = \theta^T x + b$: Prediction for linear classifiers.

Scoring losses refer to losses based on the direct evaluation of the classifier's score. In the context of linear classifiers, these scores can be greater than, less than, or equal to zero. The classification is determined based on the sign of the score, with classes labeled as either -1 or +1.

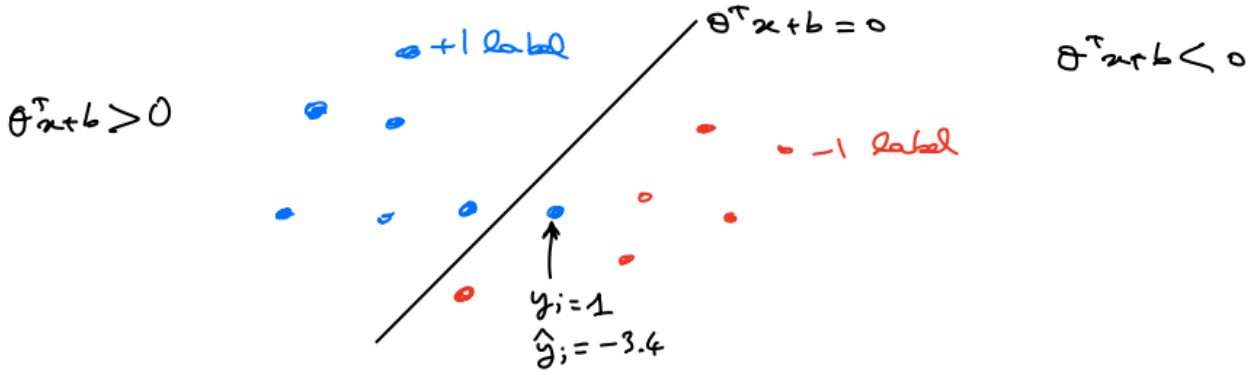


FIGURE 14 – Visualization of a decision boundary for a binary classification problem.

A significant characteristic of scoring losses is their representation as ϕ-losses or margin losses. They can generally be expressed as : $l(\hat{y}, y) = \phi(\hat{y} \times y)$, where ϕ is a certain function. This representation benefits from the binary class labels, ensuring the product does not nullify.

Examples :

1. 0/1 **Loss :** Defined by the equation
$$l^{0/1}(\hat{y}, y) = 1[y\hat{y} < 0].$$
2. **Hinge Loss :** Given by
$$l^{\text{hinge}}(\hat{y}, y) = \max(1 - y\hat{y}, 0).$$
3. **Logistic Loss :** Expressed as
$$l^{\text{logistic}}(\hat{y}, y) = \ln(1 + e^{-y\hat{y}}).$$

Many of these loss functions serve as upper bounds for the 0/1 loss, as depicted in Fig. 15.

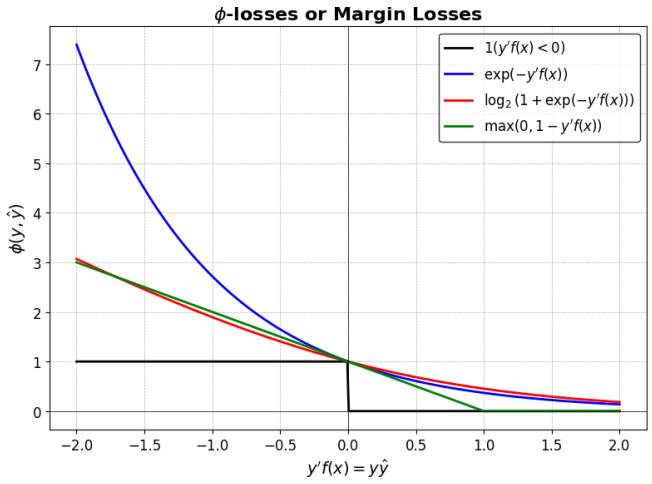


FIGURE 15 – Visualization of different Margin losses

O-1 Loss : the objective in classification tasks is often to minimize the $0-1$ loss, ensuring $\hat{y}$ aligns with y's sign.

Logistic Loss : the goal is to maximize $y\hat{y}$. Logistic regression. It aims to distance the positive class data points from the separating hyperplane as much as possible, and vice versa for the negative class. This large margin approach is more explicitly implemented by Support Vector Machines (SVMs).

Hinge Loss : Linearly penalizes misclassified points up to a margin threshold. Once a data point's margin surpasses a threshold (e.g., 1), the loss becomes zero. Consequently, hinge loss-driven classifiers work towards ensuring data points lie outside a safety margin, aiming to minimize the number of points within this margin.

In the figure below, the relationship between margin loss functions and the margin scores for points in categories (A), (B), and (C) is elucidated :

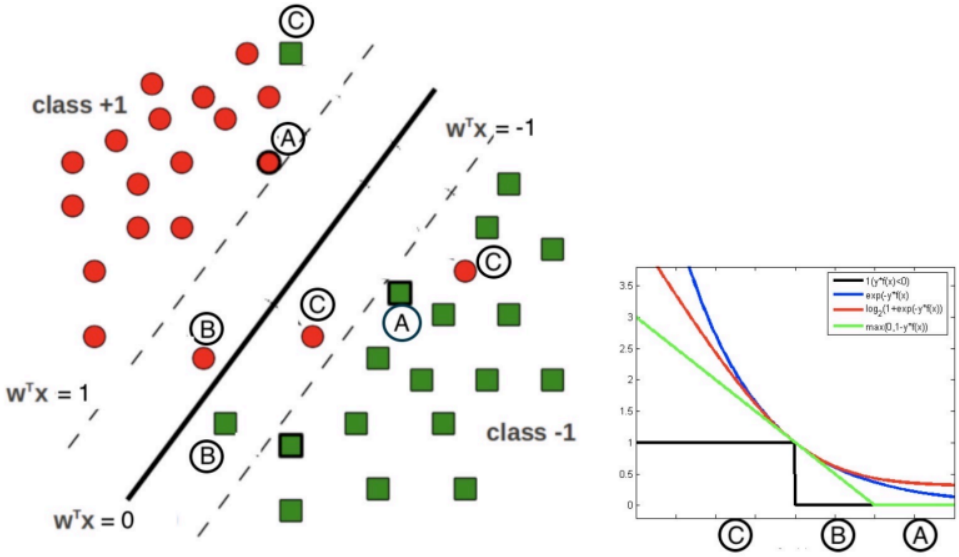


FIGURE 16 – The relationship between margin loss functions and margin scores

— **Category of points (A) :** represents correctly-classified points.Hinge loss is zero, and logistic loss is very low.

— **Category of points (B) :** represents points at the margin that are correctly classified. For a score less than one but close, both hinge and logistic losses increase.

— **Category of points (C) :** represents incorrectly-classified points. Both losses increase for scores that lie further away from the desired margin, but their rate of increase differs. The logistic loss increases exponentially, whereas the hinge loss increases linearly.

In sum, both logistic and hinge losses drive classifiers towards similar behaviors, especially when it comes to distancing data points from the decision boundary. They aim for a large margin classification, ensuring the robustness of the model.

10.1 The risk of a scoring loss

Notations :

- $\eta(x) = P(Y = 1 \mid X = x)$
- $\hat{y} \in \mathcal{R} \cup \{-\infty, \infty\}$ is a score
- $\hat{y}(.) \in \mathcal{X} \mapsto \mathcal{R} \cup \{-\infty, \infty\}$ is a scoring function (e.g. neural net).

Definition 4. For any $\hat{y} \in \mathcal{R}$, the conditional risk for scoring loss l is :

$$\mathcal{C}(\hat{y}, \eta) = \mathbb{E}_{y\sim\begin{cases} 1 & w \cdot p \cdot \quad \eta \\ -1 & w \cdot p \cdot \quad 1-\eta \end{cases}} [l(\hat{y}, y)]$$

$$= \eta l(\hat{y}, 1) + (1-\eta) l(\hat{y}, -1)$$

Definition 5. The risk of scoring function $\hat{y}(.) : \mathcal{R}^d \to \mathcal{R}$ for loss l is :

$$\mathcal{R}^l(\hat{y}(.)) = \mathbb{E}_{x,y\sim p}[l(\hat{y}(x), y)] = \mathbb{E}_x C(\hat{y}(x), \eta(x))$$

10.2 Calibration of scoring losses

What are "good" scoring losses ?

Definition 6. A loss l is calibrated if the following holds :

- if $\eta \in \left[0, \frac{1}{2}\right[$ then $\inf_{\hat{y}<0} \mathcal{C}(\hat{y}, \eta) < \inf_{\hat{y}\geq 0} \mathcal{C}(\hat{y}, \eta)$

- if $\eta \in]\frac{1}{2}, 1]$ then $\inf f_{\hat{y}>0}\mathcal{C}(\hat{y}, \eta) < \inf_{\hat{y}\leq 0}\mathcal{C}(\hat{y}, \eta)$ Theorem 1. If a loss function l is calibrated, and if $\hat{y}(.) is the measurable function that$ minimizes $\mathcal{R}^l$, then it also minimizes $\mathcal{R}^{l^{0,1}}$

Intuitively : A measurable function learnt to minimize a calibrated loss will also minimize the 0/1 loss.

Theorem 2. Any convex ϕ-loss with $\phi < 0$ is well calibrated.

Hence, all scoring losses we saw are well calibrated.

Exercise 5 :

1. Let $l_{\text{hinge}}(\hat{y}) = \max(0, \eta - \hat{y})$.
2. We will consider 3 cases in this exercise :
   (a) $\eta < \frac{1}{2}$
   (b) $\eta = \frac{1}{2}$
   (c) $\eta > \frac{1}{2}$
3. Draw $C(\hat{y}, \eta)$ as a function of $\hat{y}$ in each of these three cases.
4. In each case, show which value of $\hat{y}$ minimizes $C(\hat{y}, \eta)$.
5. Also show which predicted class corresponds to those $\hat{y}$.
6. If, instead of the hinge loss, we used the 0/1 loss, which class would be the optimal one in these 3 cases ?
7. Using the definition of calibration, show the hinge loss is calibrated.

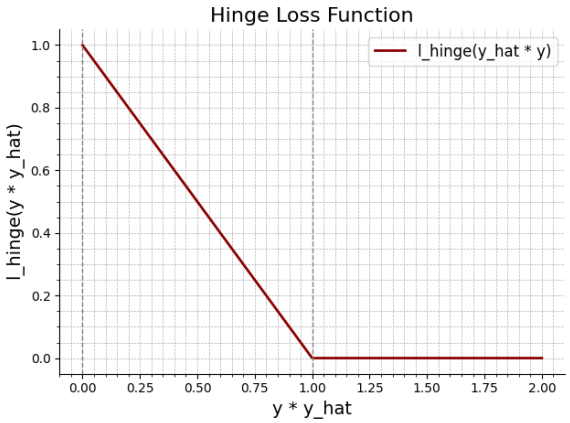


FIGURE 17 – Visualization of the Hinge loss