Fondamentaux de l'Apprentissage Automatique

Lecturer: Yann Chevaleyre
Scribe: Medjaouri Insaf

Lecture n°7 #
09/11/2023

1 Introduction

In this lecture, we will explore the fundamental concepts of linear discrimination and delve into Support Vector Machines (SVM). We will start with an investigation into the notion of Linear Separability and the strategies for Separator and Margin Maximization.Our exploration extends to the Perceptron Algorithm, examining its role as an Online Learner.After that we will uncover the resolution of the SVM Problem through the exploration of its Lagrangian. The Dual Problem, Support Vectors, and the practical aspects of SVM for both linearly separable and non-separable data will be thoroughly examined.Finally, we bridge the theoretical concepts with real-world applications, illustrating how SVM can be applied to practical problems.

2 Linear Discrimination

Linear discrimination refers to the method of classifying data points into different categories or classes based on linear combinations of features. In the context of machine learning,it involves drawing a linear boundary that separates data points of different classes as distinctly as possible.

2.1 Problem Formulation and Objective Definition

The objective is to construct a model that can classify points in a given dataset. The dataset, denoted as D, consists of points (x_i, y_i) in the input space $\mathcal{X}$ and their corresponding labels from the set {-1, 1}. The input space is assumed to be $\mathbb{R}^d$, where d represents the dimensionality of the feature space.

A decision function $f : \mathbb{R}^d \to \mathbb{R}$ is constructed using the dataset D to predict the class of a new point x. The prediction is made as follows :

— If $f(x) < 0$, then x is assigned to class -1.
— If $f(x) > 0$, then x is assigned to class 1.

The decision function is linear, expressed as $f(x) = w^T x + b$, where w is the weight vector and b is the bias, both of which are parameters to be learned from D. The function f maps the input vectors to real numbers whose sign determines the class assignment.

2.2 Linear Separability

The set of points $\{(x_i, y_i)\}$ is considered linearly separable if a hyperplane exists that can correctly discriminate the entire dataset. If such a hyperplane does not exist, the points are referred to as not linearly separable.Once the concept of linear separability is established, the next step is to determine the best linear function that separates the classes.

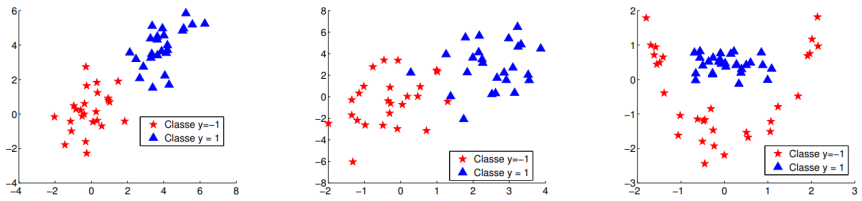

FIGURE 1 – Examples of linearly and non-linearly separable points

2.3 Separator and Margin Maximization

Finding a decision boundary is a key aspect, typically defined by the equation $w^T x + b = 0$ .However, when data is linearly separable, there can be multiple potential decision boundaries.The question then arises : Are all decision functions equal ? The answer lies in the concept of margin. The optimal decision function is one that maximizes this margin, thereby creating the widest possible gap between classes.

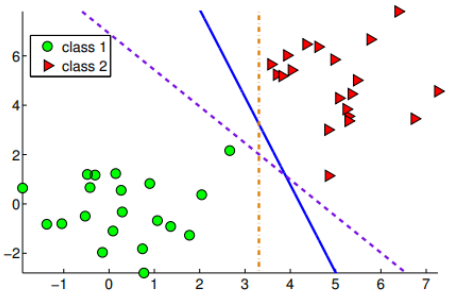

FIGURE 2 – Multiple decision boundaries and the principle of margin maximization.

In this figure, various potential linear separators are illustrated. The chosen solution should be the one that not only separates the points of classes 1 and 2 but also maximizes the margin, leading to a more robust model.

Definition 1 (Distance to the Decision Boundary)**.**
Let $H(w,b) = \{z \in \mathbb{R}^d | f(z) = w^T z + b = 0\}$ be a hyperplane and let $x \in \mathbb{R}^d$. The distance of the point x to the hyperplane H is given by

$$d(x,H) = \frac{|w^T x + b|}{\|w\|},$$

which is the absolute value of the functional margin scaled by the norm of the weight vector.

Démonstration. Let x_p be the orthogonal projection of x onto H. We have $x = x_p + d\frac{w}{\|w\|}$ which implies $d\frac{w}{\|w\|} = x - x_p$.

Taking the dot product we get $dw^T \frac{w}{\|w\|} = w^T x - w^T x_p$.

From this we deduce :

$$d\frac{\|w\|^2}{\|w\|} = w^T x + b - \underbrace{(w^T x_p + b)}_{=0},$$

And therefore :

$$d = \frac{w^T x + b}{\|w\|}.$$

□

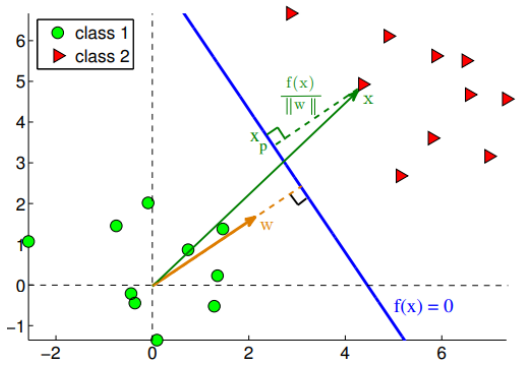

FIGURE 3 – Illustration of the distance of a point to the decision boundary.

Definition 2 (Canonical Hyperplane)**.**
A hyperplane is said to be canonical relative to the dataset $\{x_1, \ldots, x_N\}$ *if it satisfies the condition that the minimum absolute value of the functional margin,* $|w^T x_i + b|$*, for all data points* x_i *is equal to 1. Formally, for the canonical hyperplane* $H(w, b)$*, we have :*

$$\min_{x_i} |w^T x_i + b| = 1.$$

The geometric margin M of a hyperplane is a critical concept in SVM and is defined as the distance from the hyperplane to the closest data point in the dataset. For a canonical hyperplane, the geometric margin is given by :

$$M = \frac{2}{\|w\|}.$$

This definition leads us to the notion of an optimal canonical hyperplane which maximizes the margin $M$and ensures that each data point is classified correctly, which can be denoted as :

$$\forall i, \quad y_i(f(x_i)) \geq 1.$$

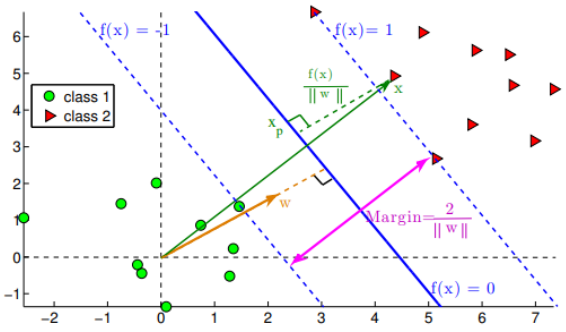

FIGURE 4 – The concept of margin in SVM.

2.4 Margin and Generalization Bound

The generalization ability of a classifier is critically dependent on the concept of the margin.The Vapnik-Chervonenkis (VC) dimension, provides a theoretical measure of the classifier's capacity to generalize to unseen data. The generalization bound is given by the following inequality :

$$R(h) \leq R_{emp}(h) + C\sqrt{\frac{D(\log(2N/D) + 1) + \log(4/\delta)}{N}}$$

where $R_{emp}(h)$ is the empirical risk, N is the sample size, D is the VC dimension of the hypothesis class $\mathcal{H}$, C is a constant, and δ is the probability with which this bound holds.

The VC dimension of linear classifiers that achieve a margin ρ can be bounded from above. For the hypothesis class $\mathcal{H}$ of functions defined by $f(x) = w^T x + b$, with a margin ρ on the training examples, the VC dimension D is bounded by :

$$D \leq 1 + \min\left(d, \frac{R^2}{\rho^2}\right)$$

Where R is the radius of the smallest sphere that encloses the training data.

2.4.1 Formulation of the Margin Maximization Problem

As we noticed the objective in SVM is to find a decision function $f(x) = w^T x + b$ that correctly discriminates all points in the dataset while also maximizing the margin. The dataset $\mathcal{D} = \{(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}\}_{i=1}^n$ consists of points that are linearly separable.

We seek to solve the following optimization problem :

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

subject to the constraints :

$$y_i(w^T x_i + b) \geq 1, \quad \forall i = 1, \ldots, n.$$

Maximizing the margin ensures that all points are correctly classified,max $\rho = \frac{1}{\|w\|}$which is equivalent to minimizing $\|w\|$.

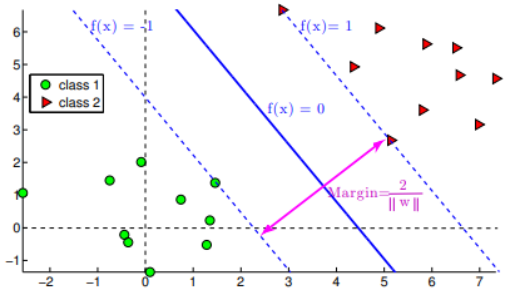

FIGURE 5 – Visualization of the SVM margin maximization problem.

2.5 The Perceptron Algorithm as an Online Learner

Moving forward from the robust framework of SVM, we now turn our attention to the Perceptron algorithm, a foundational pillar in the field of online learning for binary classification. The perceptron serves as a simple yet effective linear classifier that iteratively adjusts its weights to separate two classes.

2.5.1 Perceptron Update Rule

At each step, the algorithm receives an input vector x_t, computes a prediction $\hat{y}_t$ using the current weight vector w_t, and compares this prediction to the actual label y_t. If the prediction is incorrect, the weight vector is updated by adding the input vector scaled by the label :

Initialization : Set $t = 0$ and $w_0 = 0$.
The algorithm proceeds as follows :

— Receive an input x_t.
— Predict $\hat{y}_t = sign(w_t^\top x_t)$.
— Receive the correct label $y_t \in \{-1, 1\}$.
— If $\hat{y}_t \neq y_t$, then update $w_{t+1} = w_t + y_t x_t$.
— Otherwise, set $w_{t+1} = w_t$.
It should be noted that this process is for a homogeneous linear classifier where $f(x) = w^T \cdot x$.

2.5.2 Convergence of the Perceptron

A remarkable result due to Block and Novikoff guarantees that if the data is linearly separable, the perceptron algorithm will converge to a separating hyperplane in a finite number of updates.

Theorem 1. *Assuming that all feature vectors x have a norm less than R,$y_t \in \{-1, 1\}$and that there exists a canonical hyperplane perfectly classifying the data with a margin ρ, the number of mistakes the perceptron makes is bounded by $(R/\rho)^2$.*

Démonstration. Consider the Perceptron algorithm where the weight vector is updated upon each misclassification.

Step 1 : After an update (a prediction error), the new weight vector w_{t+1} is more aligned with w^* :

$$\langle w_{t+1}, w^* \rangle = \langle w_t + y_t x_t, w^* \rangle = \langle w_t, w^* \rangle + y_t \langle x_t, w^* \rangle \geq \langle w_t, w^* \rangle + 1.$$

This is based on the assumption that w^* is a canonical hyperplane classifying the data perfectly, and $y_t \langle x_t, w^* \rangle \geq 1$, because w^* is canonical.We get $\langle w_t, w^* \rangle \geq t_e$ which represents number of erros.

Step 2 : After an update (classification error) :
$\|w_{t+1}\|^2 = \langle w_t + y_t x_t, w_t + y_t x_t \rangle = \|w_t\|^2 + 2 y_t \langle w_t, x_t \rangle + \|y_t x_t\|^2$

$$\leq \|w_t\|^2 + R^2 \Rightarrow \|w_t\|^2 \leq t_e \cdot R^2$$

Step 3 : Combining the results of the previous steps, we can deduce :

$$t_e \leq \langle w_t, w^* \rangle \leq \|w_t\| \cdot \|w^*\| \leq \sqrt{t_e} \cdot R \cdot \|w^*\| \Rightarrow \sqrt{t_e} \leq \frac{R}{\rho}$$

$$t_e \leq \frac{R^2}{\rho^2}.$$

This completes the proof, showing that the number of mistakes t_e is bounded by $\frac{R^2}{\rho^2}$. □

2.5.3 The Perceptron Algorithm as a SGD Online Learner

The Perceptron update :
if $\mathrm{y}_t \langle w_t, x_t \rangle < 0$
$w_{t+1} \leftarrow w_t + y_t x_t$
$else$
$w_{t+1} \leftarrow w_t$

$\Delta_t = w_t^\mathsf{T} x_t$
$if y_t \Delta_t < 0$
$w_{t+1} \leftarrow w_t + y_t x_t$
$else$
$w_{t+1} \leftarrow w_t$

And the SGD Update is formulated as follow $\mathrm{w}_{t+1} \leftarrow w_t - \alpha \nabla_w L(w_t^T x_t, y)$

Perceptron Loss Function : $\ell_{perceptron}(\Delta_t, y) = \{\, 0 \; if y_t \Delta_t \geq 0; -y_t \Delta_t \quad otherwise$

Applying SGD

if $\alpha = 1$

$w_{t+1} \leftarrow w_t - \alpha \{ 0 \; if y_t \Delta_t \geq 0; -y_t x_t \quad otherwise$

3 Resolution of the SVM Problem

The resolution of the Support Vector Machine (SVM) problem involves converting a constrained optimization problem into a form that is easier to solve. This is achieved by introducing the Lagrangian, which incorporates both the objective function and the constraints.

3.1 The Lagrangian of the SVM Problem

The primal problem of an SVM can be expressed as a constrained optimization problem where we seek to minimize the norm of the weight vector subject to the constraints that each data point is correctly classified with a margin.

The primal form of the SVM problem is :

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2}\|w\|^2 \quad subject \quad to \quad y_i(w^\top x_i + b) \geq 1 \quad \forall i = 1, \ldots, n.$$

To solve this problem, we introduce Lagrange multipliers $\alpha_i \geq 0$ for each constraint, which leads to the Lagrangian :

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^\top x_i + b) - 1).$$

Taking the derivative of the Lagrangian with respect to w and setting it to zero gives :

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0.$$

Similarly, the derivative with respect to b also needs to be zero :

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0.$$

These conditions lead to the dual problem, which is easier to solve and provides a way to find the optimal values for w and b that maximize the margin while correctly classifying the training data.

3.2 The Dual Problem

Having established the Lagrangian of the SVM problem, we proceed by examining the dual problem, which provides a more computationally efficient path to solving the SVM.

From the stationarity condition, we obtain the following equations :

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \quad and \quad \frac{\partial L(w, b, \alpha)}{\partial w} = 0.$$

The dual problem is a quadratic programming problem that can be formulated as follows :

$$\max_{\alpha} \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^\top x_j \right)$$

subject to the constraints :

$$\alpha_i \geq 0, \quad \forall i = 1, \ldots, n \quad and \quad \sum_{i=1}^{n} \alpha_i y_i = 0.$$

3.2.1 Support Vectors

In the context of the dual problem of SVM, solving for the Lagrange multipliers α_i provides insight into the structure of the solution. Specifically, we find that :

— For a point x_j where $y_j(w^\top x_j + b) > 1$, the corresponding $\alpha_j = 0$.

— Conversely, for the points x_i where $y_i(w^\top x_i + b) = 1$, the $\alpha_i \geq 0$.

This leads to the solution for w being expressed as :

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i,$$

Which implies that w is a linear combination of only the support vectors.The weight vector w and the bias b define the decision boundary, which can be represented as $w^\top x + b = 0$. The support vectors are the data points that satisfy $y_i(w^\top x_i + b) = 1$, lying on the margins of the dividing hyperplane.

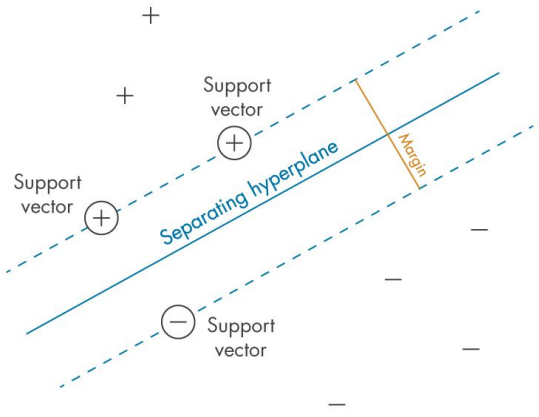

FIGURE 6 – Illustration of the support vectors.

3.2.2 Practical SVM for Linearly Separable Data

To find the weight vector w, we use the training data $D = \{(x_i, y_i)\}_{i=1}^{n}$ and the Lagrange multipliers α_i from the dual problem :

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

This equation indicates that w is a linear combination of the support vectors, which are the data points corresponding to non-zero α_i.

The bias b is determined by the support vectors which satisfy the following condition :

$$y_i(w^\top x_i + b) = 1$$

By solving this for b, we can find its value which is crucial for the placement of the decision boundary.

The decision function of the SVM is used to classify new data points and is given by :

$$f(x) = w^\top x + b = \sum_{i=1}^{n} \alpha_i y_i x_i^\top x + b$$

This function assigns a data point x to a class based on the sign of $f(x)$, effectively using the model constructed from the support vectors and their corresponding Lagrange multipliers.

3.2.3 Handling Non-Separable Cases in SVM

When dealing with non-linearly separable data, the SVM needs to adjust the constraints to allow for some misclassifications. This is achieved by introducing slack variables ξ_i, which measure the degree of misclassification of the data point x_i :

— Adjust the constraint $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$.
— Permit $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$, with $\xi_i \geq 0$ representing the error term.
— Incorporate the sum of errors $\sum_{i=1}^{n} \xi_i$ into the SVM problem.

The SVM formulation for the non-separable case includes a regularization parameter C which controls the trade-off between the slack variable penalty and the margin size. The optimization problem becomes :

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

subject to

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \forall i$$

$$\xi_i \geq 0, \quad \forall i$$

$C > 0$: regularization parameter (trade-off between error and margin). C is to be set by the user.

The dual problem incorporates the slack variables and the regularization parameter :

$$L(w, b, \xi, \alpha, \nu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i(y_i(w^\top x_i + b) - 1 + \xi_i) - \sum_{i=1}^{n} \nu_i\xi_i$$

where $\alpha_i \geq 0$ and $\nu_i \geq 0$.
For the stationarity Optimality Conditions :

$$\frac{\partial L(w, b, \xi, \alpha)}{\partial b} = 0, \quad \frac{\partial L(w, b, \xi, \alpha)}{\partial w} = 0, \quad \frac{\partial L(w, b, \xi, \alpha)}{\partial \xi_k} = 0$$

which gives us the following conditions :

$$\sum_i \alpha_i y_i = 0, \quad w = \sum_i \alpha_i y_i x_i, \quad C - \alpha_i - \nu_i = 0, \quad \forall i = 1, \ldots, n$$

The solution to the dual problem can be summarized as follows :

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j x_i^\top x_j$$

subject to

$$0 \leq \alpha_i \leq C, \quad \forall i$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Theorem 2. *Let there be a non-separable linear SVM decision problem $f(x) = w^T x + b$. The vector w is defined by $w = \sum_{i=1}^{n} \alpha_i y_i x_i$, where the coefficients α_i are solutions of the dual problem mentioned above.*
What has changed ? Nothing except the constraints on α_i, which are now $0 \leq \alpha_i \leq C$.

The parameter C has a significant impact on the SVM's solution. A smaller C results in a wider margin , whereas a larger C leads to a narrower margin.

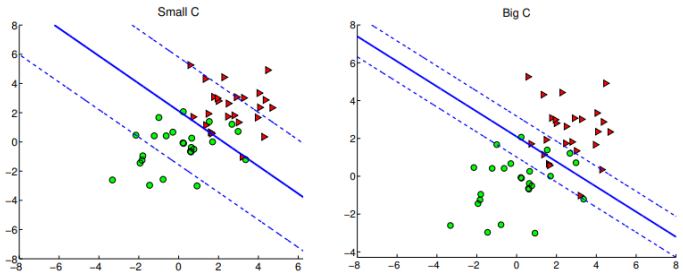

FIGURE 7 – the effect of C parameter.

4 SVM in Practice

4.1 Practical Implementation

Input Elements :
— Labeled data : $\{(X_i, y_i)\} \in \mathbb{R}^d \times \{-1, 1\}^n$
Methodology :

1. Center the data : $\{X_i\}_{i=1}^n \rightarrow \{X_i = X_i - \overline{X}\}_{i=1}^n$
2. Set the SVM parameter $C > 0$
3. Use a solver to solve the dual problem and obtain the non-zero Lagrange multipliers α_i, the corresponding support vectors X_i, and the bias b
4. Derive the decision function : $f(X) = \sum_{i \in SV} \alpha_i y_i X_i^T X + b$
5. Evaluate the generalization error of the obtained SVM (cross-validation, etc.). Restart from step 2 if the result is not satisfactory.

4.2 Parameter C Adjustment : A Practical Procedure

— Training set : to calculate w and b
— Validation set : to evaluate the error of classification for different C
— Test set : evaluation of the "best model"

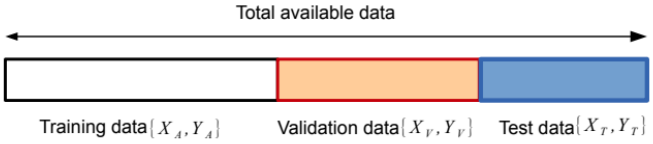

FIGURE 8 – Total available data.

Model selection : tuning of C

function $C \leftarrow tuneC(X, Y, options)$

(a) Split the data $(X_a, Y_a, X_v, Y_v) \leftarrow SplitData(X, Y, options)$
(b) For different values of C :
  i. $(w, b) \leftarrow TrainLinearSVM(X_a, Y_a, C, options)$
  ii. $error \leftarrow EvaluateError(X_v, Y_v, w, b)$
(c) Return $C \leftarrow \arg\min error$

4.3 Example

— The values of C chosen on a logarithmic scale
— For each C, we train an SVM and calculate its validation error
— The minimum of the error curve corresponds to the best value C^*
— The corresponding SVM is on the figure to the right

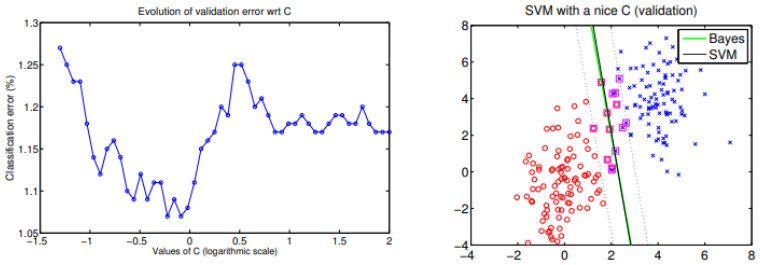

FIGURE 9 – SVM Validation Error and Optimal C Parameter Visualization.

5 Relationship between Soft-SVM, Hinge-loss, and Hinge-loss Perceptron

The soft-SVM problem with slack variables is formulated as follows :

$$\min_{\omega} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N}\xi_i \quad s.t. \quad y_i(\langle\omega, x_i\rangle + b) \geq 1 - \xi_i; \quad \xi_i \geq 0$$

Constraints on ξ_i :

$$\xi_i \geq 0; \quad \xi_i \geq 1 - y_i(\langle\omega, x_i\rangle + b) = 1 - y_i\Delta_i$$

with $\Delta_i = \langle\omega, x_i\rangle + b$

This implies :

$$\xi_i \geq \max(0, 1 - y_i\Delta_i)$$

Consider this optimization sub-problem :

$$\min_{\xi} \sum \xi_i \quad s.t. \quad \xi_i \geq \max(0, 1 - y_i\Delta_i)$$

Solution :

$$\xi_i = \max(0, 1 - y_i\Delta_i)$$

The soft-SVM problem becomes :

$$\min_{\omega} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N}\max(0, 1 - y_i(\langle\omega, x_i\rangle + b))$$

Which reduces to :

$$\min_{\omega} \frac{1}{2C}||\omega||^2 + \sum \ell_{hinge}(\langle\omega, x_i\rangle + b, y_i)$$

where $\ell_{hinge}(\Delta_i, y_i) = \max(0, 1 - y_i\Delta_i)$

And the perceptron-loss is defined as :

$$\ell_{perceptron}(\Delta_t, y) = \max(0, -y\Delta_t)$$

SGD on the objective function :

$$\nabla_w \left(\frac{1}{2C}||w||^2 + \sum_i \ell(\Delta_i, \hat{y}_i) \right) = \frac{w}{C} + \sum_i \begin{cases} 0 & if \quad y_i\delta_i > 1 \\ -y_i x_i & else \end{cases}$$

SGD update rules for Soft SVM are as follows :

$$\omega_{t+1} = \begin{cases} \omega_t + \alpha y_t x_t - \frac{\alpha}{C}\omega_t & if \quad y_t\langle \omega_t, x_t \rangle < 1 \\ \omega_t - \frac{\alpha}{C}\omega_t & else \end{cases}$$

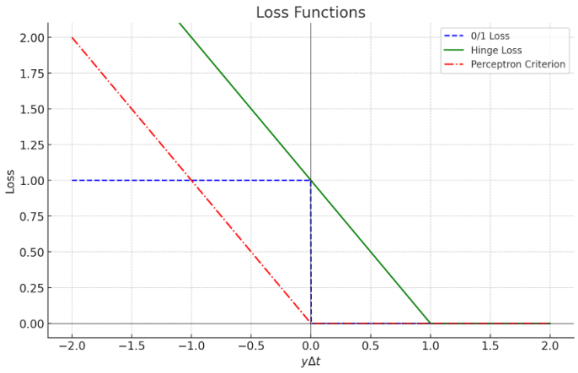

FIGURE 10 – Comparative Visualization of Different Loss Functions.