Understanding the Loss Function

Concrete illustration in Linear Regression

Kheireddin Kadri

DVRC

October 20, 2025

Outline – Part 1: Loss Functions

Why a Loss Function?

- In machine learning, a model learns by **minimizing an error**.
- This error is measured through a **loss function**.
- It expresses the difference between the **predicted value** and the **true value**.

Intuitive Theorem

The lower the loss, the better the model captures the underlying structure of the data.

Formal Definition

Let $f_\theta(x)$ be a model with parameters θ and a data point (x_i, y_i).

The loss function L is defined as:

$$L(y_i, \hat{y}_i) = L(y_i, f_\theta(x_i))$$

The global objective is:

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_\theta(x_i))$$

Remark

This is the quantity that the gradient descent algorithm seeks to minimize.

Concrete Example

We want to predict a person's height y based on their age x:

$$\hat{y} = wx + b$$

The loss function used is the **Mean Squared Error (MSE)**:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Numerical Example

x_i	y_i (true)	$\hat{y}_i$ (predicted)
10	140	145
15	160	155
20	170	172

$$L = \frac{(140-145)^2 + (160-155)^2 + (170-172)^2}{3} = 14.67$$

Loss Visualization

- The MSE creates a convex surface: there is a single global minimum.
- The gradient moves toward this minimum by adjusting w and b.

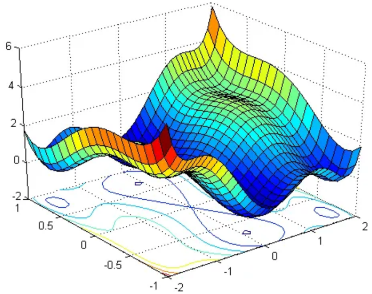

Figure: Surface of the MSE loss function with respect to w and b Source : https://ics.uci.edu/ xhx/courses/CS206/

Varieties of Loss Functions

Classification

- Cross-Entropy Loss
- Hinge Loss (SVM)

Regression

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- Huber Loss

Strategic Choice

The choice depends on the **type of problem** and the **desired robustness**.

Summary

- The loss function measures the distance between the model and reality.
- Its choice directly affects the stability and speed of learning.
- Optimization consists in **descending along the gradient** of this function.

Key Message

To understand the loss function is to understand the heart of learning!

Thank you for your attention !

Any questions?

The K-Nearest Neighbors (KNN) Algorithm

Understanding, Implementing, Interpreting

Kheireddin Kadri

DVRC

October 20, 2025

Outline – Part 2: KNN Algorithm

Basic Idea

- KNN is a **supervised and non-parametric** algorithm.
- It classifies a point according to the **K closest examples**.
- The decision is based on a **majority vote**.

Intuitive Theorem

Structural similarity in the feature space determines the class of a new sample.

Distance Measure

For two points x_i and x_j:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

(Euclidean distance)

Alternatives

- Manhattan distance: $\sum |x_{ik} - x_{jk}|$
- Minkowski distance: $(\sum |x_{ik} - x_{jk}|^p)^{1/p}$

Example: Iris Flowers

- Data: petal and sepal length and width.
- Objective: predict the species of a flower.

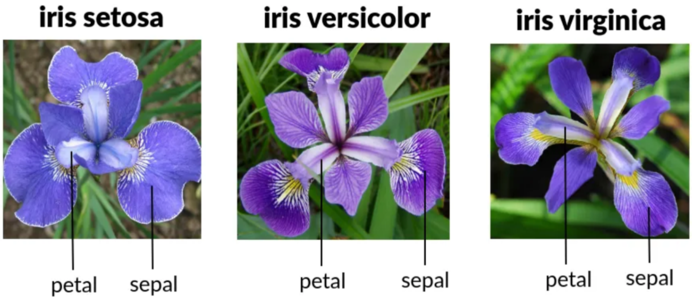

Figure: Decision boundaries of the KNN model on the Iris dataset.

Example: Iris Flowers (part2)

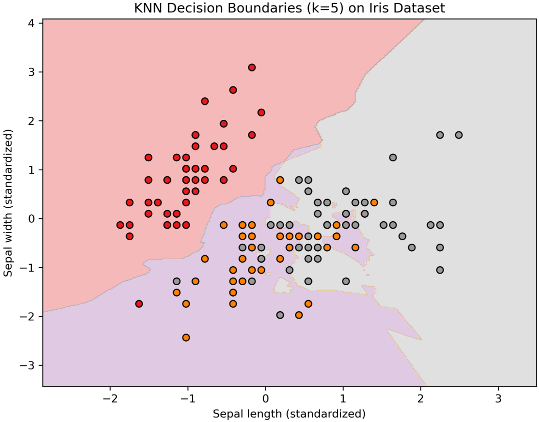

Figure: Decision boundaries of the KNN model on the Iris dataset.

Influence of the Parameter k

- Small k: model too sensitive to noise (overfitting).
- Large k: model too smooth (underfitting).

Good Balance

Choose k that minimizes the error on a cross-validation set.

Strengths and Weaknesses

Advantages

- Simple to understand and implement.
- No explicit training phase.
- Performs well on small datasets.

Disadvantages

- High computational cost for large-scale data.
- Sensitive to normalization and high dimensionality.

Summary

- KNN relies on proximity to classify or predict.
- Key parameters: k, distance metric, normalization.
- Useful as a first simple and intuitive ML approach.

Key Message Takeaway

Proximity is a form of learning by mapping: to understand is to recognize resemblance.

Thank you for your attention !

Any questions?

Understanding Decision Trees

From Theory to Practice

Kheireddin Kadri

DVRC

October 20, 2025

Outline – Plan Part 3 : DT Algorithm

Concept

Definition

A Decision Tree is a flowchart-like model that recursively splits data into subsets based on feature values to make predictions.

- Each node represents a feature condition.
- Each branch represents an outcome.
- Each leaf represents a class label.

Entropy

Formula

$$Entropy(S) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

- Measures the impurity of a set.
- Entropy $= 0 \rightarrow$ Pure subset.

Information Gain

Definition

Information Gain measures the reduction in entropy after a dataset is split:

$$IG(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Goal

Choose the feature that maximizes the Information Gain at each split.

Overfitting

- Deep trees may fit training data too closely.
- Performance drops on unseen data.

Solution

- Limit tree depth (‘max$_d$*epth*‘)
- Set minimum samples per leaf (‘min$_s$*amples$_l$eaf*‘)
- Use pruning techniques

Example: Iris Dataset

- Load the Iris dataset
- Train a Decision Tree with `criterion='entropy'`
- Visualize the tree and decision boundaries
- Evaluate model accuracy

Result

Decision Trees can perfectly separate classes in simple 2D projections.

Example: Iris Flowers (part3)

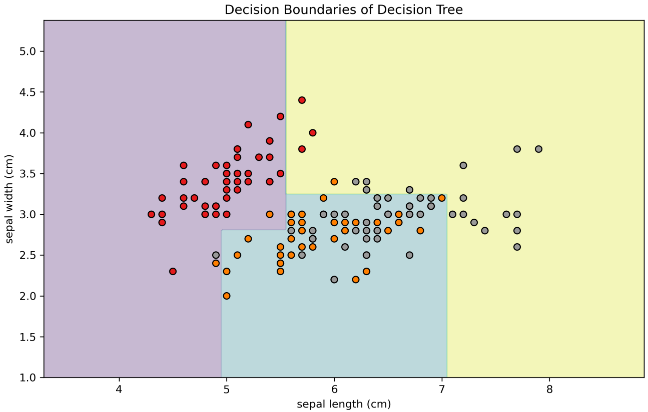

Figure: Decision boundaries of the DT model on the Iris dataset.

GridSearchCV

Parameters to optimize

- `criterion`: 'gini' or 'entropy'
- `max_depth`: limits tree growth
- `min_samples_split`, `min_samples_leaf`

Objective

Automatically select the best combination that yields the highest cross-validation score.

Summary

- Decision Trees split data using entropy and information gain.
- Simple to interpret and visualize.
- Prone to overfitting — pruning is essential.
- Stronger versions: Random Forest, Gradient Boosted Trees.

Support Vector Machines (SVM)

From Linear to Nonlinear Classification

Kheireddin Kadri

DVRC

October 20, 2025

Outline – Plan Part 4 : SVM Algorithm

Why SVM?

Problem

We need a classifier that can separate data points into two or more classes with the largest possible margin.

Key Idea

SVM finds the hyperplane that maximizes the margin between classes.

The Hyperplane Concept

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

The goal is to find $\mathbf{w}$ and b that separate classes with:

$$\text{maximize } \frac{2}{\|\mathbf{w}\|}$$

subject to correct classification constraints.

Support Vectors

- Points closest to the decision boundary.
- Define the margin.
- Only these points influence the model.

The Kernel Trick

Idea

Transform data to a higher-dimensional space where it becomes linearly separable.

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

- Linear kernel: $K(x, y) = x \cdot y$
- Polynomial kernel: $(x \cdot y + 1)^d$
- RBF kernel: $\exp(-\gamma \|x - y\|^2)$

Model Evaluation

- Confusion Matrix
- Accuracy, Precision, Recall, F1-score
- Cross-validation

Advantages and Limitations

Advantages

- Works well with high-dimensional data
- Effective with clear margin of separation

Limitations

- Memory and computation heavy for large datasets
- Requires careful tuning of kernel parameters

Conclusion

- SVM maximizes margin $\rightarrow$ robust decision boundary
- Kernel trick $\rightarrow$ nonlinear classification
- Always validate with cross-validation