

CSE 5819 - Assignment 6

Riley Francis - rif17002

Assignment_6.pdf



Part 2

Support Vector Machine

a.

We want to find the Lagrangian of the formulation

$$\min_{w, b, \xi} \left\{ \frac{1}{2} w^\top w + \frac{C}{2} \sum \xi_i^2 \right\}$$

subject to $y_i(w^\top w + b) \geq 1 - \xi_i$ and $i \in \{1, \dots, N\}$.

We can rearrange the constraint to get $1 - \xi_i - y_i(w^\top w + b) \leq 0$. If we let $g_i(w, b, \xi) = 1 - \xi_i - y_i(w^\top w + b)$, then this becomes $g_i(w, b, \xi) \leq 0$. We are trying to find a Lagrangian:

$$\mathcal{L} = \text{objective} + \sum_i \alpha_i g_i$$

The objective comes directly from the first equation above. If we introduce the Lagrange multiplier α , the Lagrangian of the formula above is then

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

b.

The partial derivative of the Lagrangian from [part a](#) with respect to w , b , and ξ is as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w} &= w - \sum_{i=1}^N \alpha_i y_i x_i = 0 \Rightarrow & w &= \sum_{i=1}^N \alpha_i y_i x_i \\ \frac{\partial \mathcal{L}}{\partial b} &= \sum_{i=1}^N \alpha_i y_i = 0 \Rightarrow & 0 &= \sum_{i=1}^N \alpha_i y_i \\ \frac{\partial \mathcal{L}}{\partial \xi_i} &= C \xi_i - \alpha_i = 0 \Rightarrow & \xi_i &= \frac{\alpha_i}{C}\end{aligned}$$

Decision Trees

a.

First we calculate the Gini impurity of y :

$$Gini(s) = 1 - \sum_i p_i^2 = 1 - (p_{positive}^2 + p_{negative}^2) = 1 - ((0.666)^2 + (0.333)^2) = 0.445$$

Next, we find the Gini impurity of each of the possible splits.

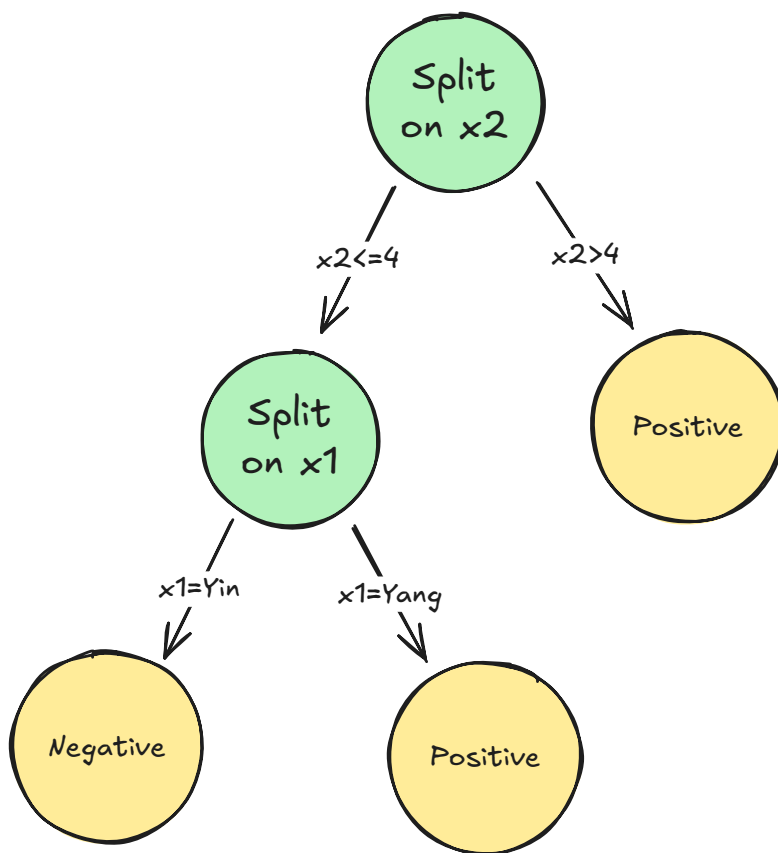
For x_1 , we have $\frac{4}{6}(1 - (0.5^2 + 0.5^2)) + \frac{2}{6}(1 - 1^2 + 0^2) = 0.333$, so the information gain is 0.112.

Next, we calculate the Gini impurity of each of the possible splits of x_2 :

Threshold	Weighted Gini	Gini Gain
1	0.445	0.000
3	0.417	0.028
4	0.222	0.223
6	0.333	0.112
7	0.267	0.178
11	0.400	0.045

Since the largest Gini gain is x_2 with a threshold of $x_2 \leq 4$, that is the split that we will use.

b.



C.

The root node of the decision tree could not remain the same since this would lead to wrongly classifying the altered data point as positive. We can see this by recalculating the Gini table:

Splitting on x_1 : Gini gain = 0.25

Splitting on x_2 :

Threshold (x_2)	Weighted Gini	Gini Gain
1	0.500	0.000
3	0.444	0.056
4	0.333	0.167
6	0.333	0.167
7	0.333	0.167
11	0.444	0.056

As you can see, the new largest Gini gain is actually splitting on x_1 instead.

Part 3: Programming

[Submitted separately](#)

Part 1: ChatGPT

1. What is the most used evaluation metrics for classification?

The most commonly used **evaluation metrics for classification** depend on whether the problem is **binary**, **multiclass**, or **imbalanced**, but here are the main ones used across contexts:

1. Accuracy

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

- **Most used** for balanced datasets.
- **Weakness:** Misleading when classes are imbalanced (e.g., 95% of one class → 95% accuracy by predicting only that class).

2. What is the most used evaluation metrics for regression?

The most commonly used **evaluation metrics for regression** measure how well a model's continuous predictions match true target values. Here are the main ones — grouped by type and popularity:

Most Commonly Used

1. Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- **Intuition:** Penalizes large errors more heavily (since errors are squared).
- **Common in:** Linear regression, neural networks, and most optimization objectives.

3. For supervised learning algorithms, what would be a reliable procedure

of evaluating their performance?

Excellent question — this is central to building **trustworthy supervised learning models**. Here's a structured and reliable **procedure for evaluating the performance of *any supervised learning algorithm*** (classification or regression):

Step 1: Split Your Data Properly

A solid evaluation starts with **data partitioning** to ensure generalization.

Typical Splits

- **Training set** → used to fit the model.
- **Validation set** → used for model selection / hyperparameter tuning.
- **Test set** → used *only once* for final performance evaluation.

Common ratios:

- **70% / 15% / 15%** (train/val/test) or
- **80% / 20%** (train/test) if using **cross-validation** for tuning.

4. Study bagging (bootstrapping aggregation)?

Absolutely — let's **study bagging (Bootstrap Aggregating)** carefully from the ground up. We'll go through what it is, why it works, how it's used, and even peek at an example of its implementation.

1. What Is Bagging?

Bagging stands for **Bootstrap Aggregating**, an ensemble technique designed to **reduce variance** and **increase model stability**.

♦ Key Idea:

Train many versions of a model on slightly different datasets, then **average or vote** their predictions.

5. Why the bagging procedure can create models of lower variance?

Great question — this goes right to the **core idea behind ensemble learning!**

Let's unpack *why bagging (Bootstrap Aggregating)* can reduce **variance** in supervised learning models.

🎯 Intuition First: What "Variance" Means

In supervised learning:

- **Bias** measures how far predictions are from the *true function* on average.
- **Variance** measures how much the model's predictions **fluctuate** if you train it on different datasets.

A **high-variance model** (like an unpruned decision tree) may fit training data very closely but change drastically with small data perturbations — i.e., it overfits.

6. What are the advantages of Support vector machine?

Support Vector Machines (SVMs) have several **key advantages** that make them one of the most powerful and reliable algorithms in supervised learning—especially for classification. Here's a complete breakdown 📌

🧠 1. Effective in High-Dimensional Spaces

- SVMs perform very well when the number of **features is large** relative to the number of samples (e.g., text classification, bioinformatics).
- The decision boundary depends only on a subset of training points (the **support vectors**), not on the entire dataset—this makes them less prone to the curse of dimensionality.

7. What is support vector regression? (We have discussed support vector classifier, but SVM can also be extended to solve regression problems.)

Excellent — you're right!

Just like the **Support Vector Classifier (SVC)** handles classification, the same core idea can be extended to **regression tasks**, leading to **Support Vector Regression (SVR)**.

Let's unpack what it is and how it works 📌

🧩 1. What Is Support Vector Regression (SVR)?

Support Vector Regression (SVR) is the **regression version of SVM**.

Instead of finding a hyperplane that separates classes, SVR finds a **function that approximates the target values** with **maximum margin tolerance**.

🧠 **Goal:** Find a function $f(x)$ that deviates from the actual targets y_i by at most ϵ (epsilon), while keeping the model as flat (simple) as possible.

8. How many different formulations of SVM?

Excellent question — and a subtle one!

There are **several formulations of Support Vector Machines (SVMs)**, depending on the **type of problem**, the **data separability**, and the **loss function used**.

Let's go through all the major formulations — from the foundational ones to their key extensions 📌

🧩 1. Hard-Margin SVM

✅ **For: Linearly separable data**

- Objective: Find the **maximum-margin hyperplane** that separates the two classes *perfectly*.
- No allowance for misclassification — every point must be on the correct side of the margin.

9. How to solve an SVM optimization problem?

Excellent — this is where the mathematics behind SVMs comes alive 🔥

Let's walk through **how to solve an SVM optimization problem**, step by step — from the basic **primal form**, through its **dual formulation**, to the **solution process** used in practice (like SMO or Quadratic Programming).

🧩 1. Start with the Goal of SVM

The idea of SVM is to find the **optimal separating hyperplane**:

$$f(x) = w^T x + b$$

that maximizes the **margin** between two classes, while possibly allowing some misclassifications.