

Optimization and data analysis with Python - Gurobi

Antonia Ilabaca – Gonzalo Ríos

Doctorantes Ingeniería Industrial

Student Chapter – EII PUCV



Optimization and data analysis with Python - Gurobi

Antonia Ilabaca – Gonzalo Ríos

Doctorantes Ingeniería Industrial

Student Chapter – EII PUCV



Contenidos

Análisis y manipulación de datos con bases incorporadas en Python

Describir un conjunto de datos en términos de análisis univariado y multivariado

Entender los elementos de las máquinas de soporte vectorial (SVM)

Construir un modelo de clasificación a través de programación matemática

Evaluar el desempeño del modelo de clasificación construido

Análisis y manipulación de datos con bases incorporadas en Python

Base de datos Iris para análisis y manipulación

Descripción del conjunto de datos con funciones incorporadas en Python

Desarrollo de visualizaciones customizadas de los datos

Análisis y manipulación de datos con bases incorporadas en Python



Conjunto de datos multivariante

Contiene 50 muestras de tres especies (Setosa, Virginica y Versicolor)

Cuatro características por muestra (largo y ancho de sépalo y pétalo)

Entender los elementos de las máquinas de soporte vectorial (SVM)

Repaso de la idea detrás de los SVM para clasificación

Entender los supuestos del modelo y los elementos que componen al SVM clásico

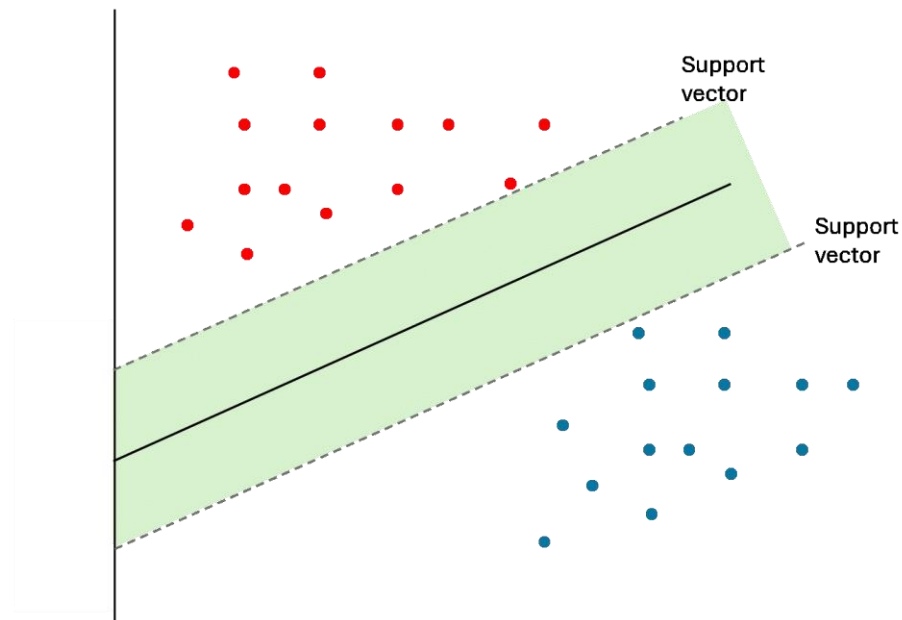
Desarrollar un modelo a través de optimización matemática para clasificar datos

Entender los elementos de las máquinas de soporte vectorial (SVM)

Problema de clasificación binaria para un tamaño de muestra N con p características

SVM es un clasificador de margen máximo, utilizando una serie de parámetros para separar datos

$$\{x_i \in \mathbb{R}^p, i = 1, \dots, N\}$$
$$\{y_i \in \{-1, +1\}, i = 1, \dots, N\}$$



Entender los elementos de las máquinas de soporte vectorial (SVM)

Modelo de clasificación SVM basa su formulación en un problema de programación cuadrática

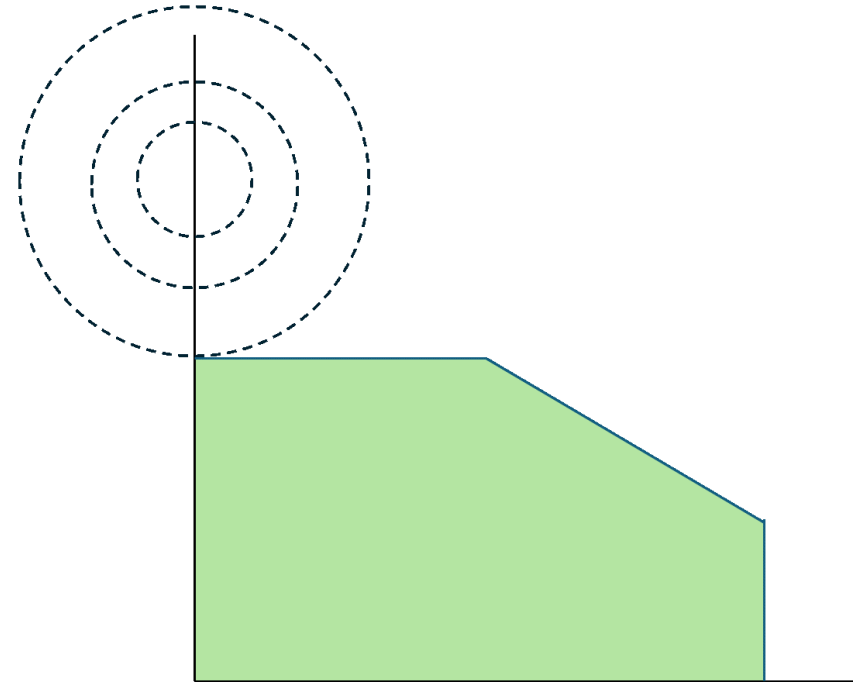
$$\min_x f_0(x) \rightarrow \min_x c^T x + x^T Q x$$

$$\text{subject to: } Ax \preceq b$$

$$Q = Q^T \succeq 0$$

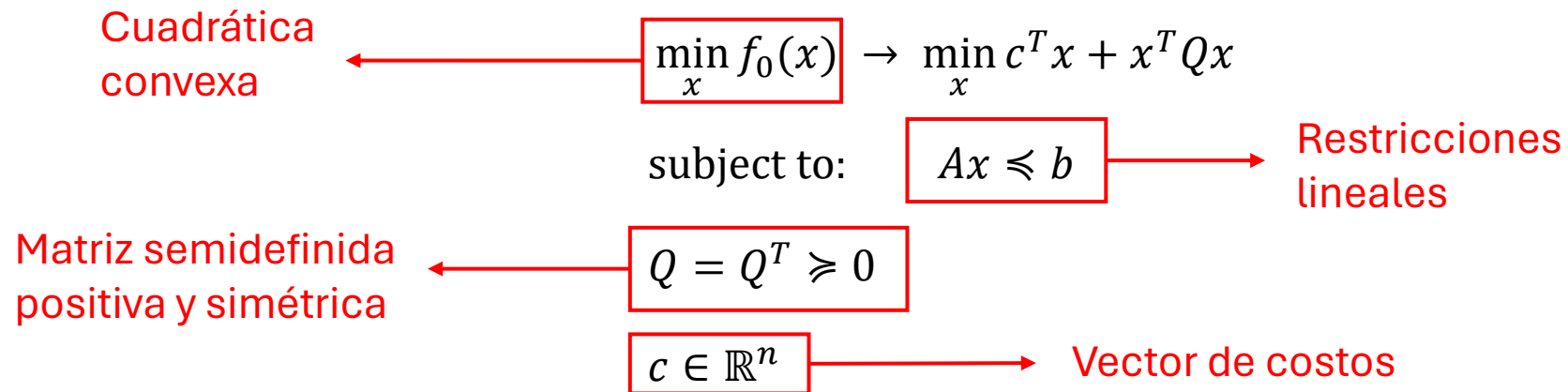
$$c \in \mathbb{R}^n$$

Programación lineal es un caso particular de programación cuadrática con la matriz Q igual a cero



Entender los elementos de las máquinas de soporte vectorial (SVM)

Modelo de clasificación SVM basa su formulación en un problema de programación cuadrática



Elementos de un problema de programación cuadrática

Entender los elementos de las máquinas de soporte vectorial (SVM)

Forma general de un modelo SVM clásico

$$\min_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^N \epsilon_i$$

subject to: $y_i(x_i^T w + b) \geq 1 - \epsilon_i,$
 $\epsilon_i \geq 0$

Donde: $w = \beta$ = pesos

$\beta_0 = b$ = sesgo

C = parámetro de costo

$||w||$ = módulo

$$||w|| = \sqrt{\sum_{i=1}^p \beta_i^2}$$

Entender los elementos de las máquinas de soporte vectorial (SVM)

Forma general de un modelo SVM clásico

Minimizamos el módulo
para maximizar el margen

$$\min_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^N \epsilon_i$$

Trade-off para casos donde los datos
no son linealmente separables

subject to:

$$y_i(x_i^T w + b) \geq 1 - \epsilon_i,$$
$$\epsilon_i \geq 0$$

Donde: $w = \beta$ = pesos

$\beta_0 = b$ = sesgo

C = parámetro de costo

$||w||$ = módulo

$$||w|| = \sqrt{\sum_{i=1}^p \beta_i^2}$$

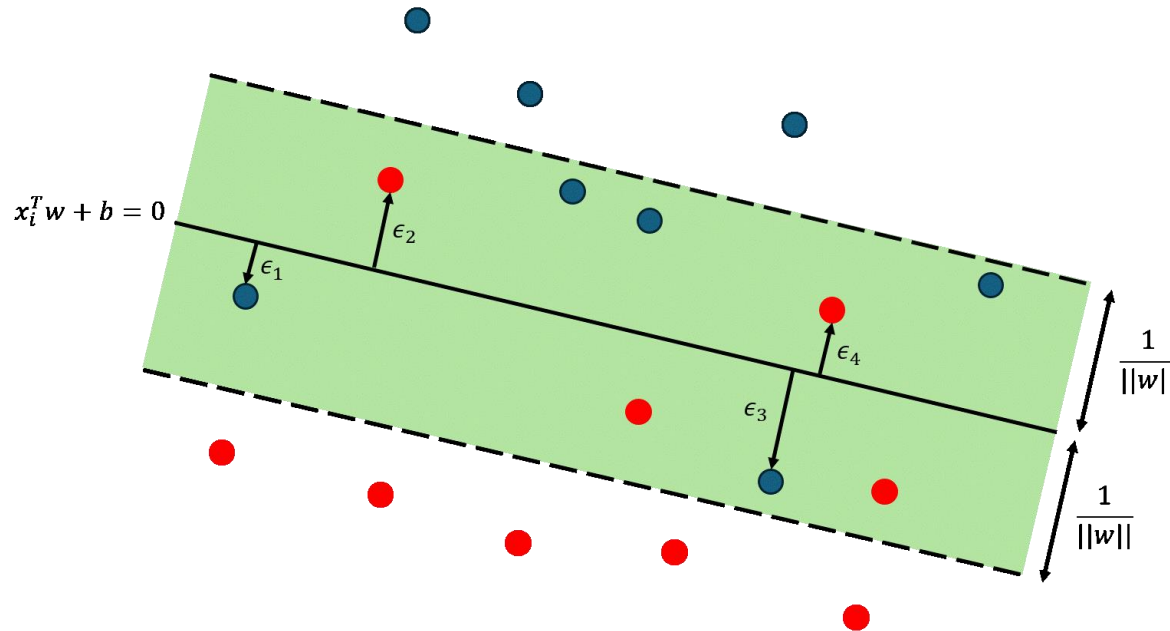
Entender los elementos de las máquinas de soporte vectorial (SVM)

Minimizamos el módulo
para maximizar el margen

$$\min_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^N \epsilon_i$$

Trade-off para casos donde los datos
no son linealmente separables

subject to: $y_i(x_i^T w + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0$



Construir un modelo de clasificación a través de programación matemática

Desarrollar el modelo computacional a través de Gurobi y Python

Definir la estructura y los elementos del SVM en Gurobi

Resolver el modelo y construir las reglas de clasificación para el etiquetado de datos

Construir un modelo de clasificación a través de programación matemática



Plataforma de análisis prescriptivo a través de optimización matemática



Herramienta de resolución de problemas de programación matemática

Implementado en Python para facilitar el desarrollo en este lenguaje

Construir un modelo de clasificación a través de programación matemática

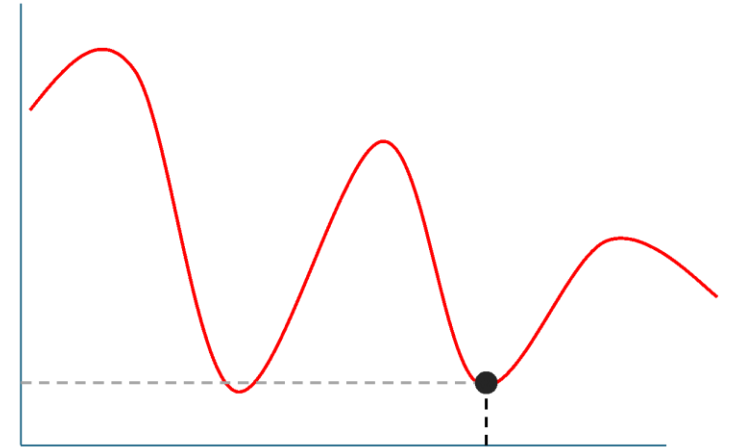


Definir variables y parámetros del modelo

Definir el modelo a través de su función objetivo

Definir restricciones del modelo

Utilizar un Solver en Gurobi para resolver el modelo generado



Evaluar el desempeño del modelo de clasificación construido

Medir el desempeño de la clasificación a través de la matriz de confusión asociada

Calcular el tiempo de resolución del problema

Visualizar la regla de clasificación

Evaluar el desempeño del modelo de clasificación construido

Matriz de confusión asociada a un problema de clasificación

| | | Predicted | |
|--------|--------------|---------------------|---------------------|
| | Population | Positive (PP) | Negative (PN) |
| Actual | Positive (P) | True Positive (TP) | False Negative (FN) |
| | Negative (N) | False Positive (FP) | True Negative (TN) |

$$TP\ Rate = \frac{TP}{P}$$

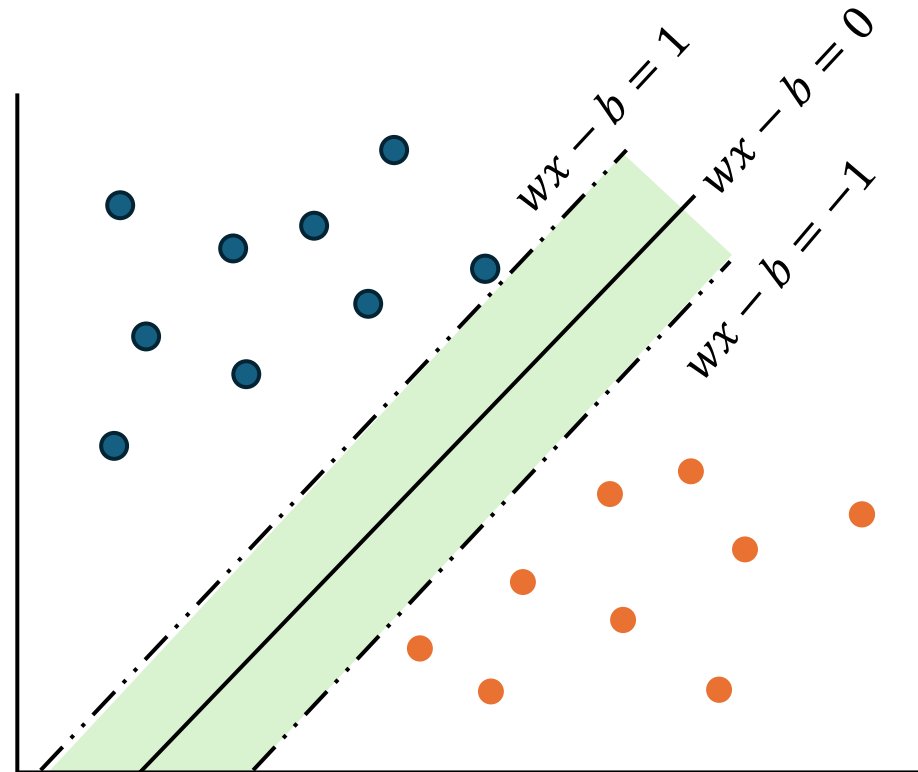
$$TN\ Rate = \frac{TN}{N}$$

$$Accuracy = \frac{TP + TN}{P + N}$$

$$F_1\ Score = \frac{2TP}{2TP + FP + FN}$$

Evaluar el desempeño del modelo de clasificación construido

Visualizar la regla de clasificación



Optimization and data analysis with Python - Gurobi

Antonia Ilabaca – Gonzalo Ríos

Doctorantes Ingeniería Industrial

Student Chapter – EII PUCV

