

1) Modelo y problema de optimización de

- SVC (Support Vector Classifier):

SVC busca encontrar el hiperplano que maximiza el margen entre 2 clases. Es decir, la distancia entre los puntos de borde (soporte) y el hiperplano.

Función objetivo (caso lineal, margen blando):

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

Sujeto a:

$$y_i (w'x_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0$$

w : Vector normal al hiperplano.

b : Sesgo.

ϵ_i : Variables de holgura (permiten errores)

C : Hiperparámetro que controla el compromiso entre margen y error.

$y_i \in [-1, 1]$ etiqueta de la clase.

x_i : Punto de entrenamiento.

Para problemas No lineales: Se utiliza un kernel (ej: RBF, polinomial), lo que transforma el problema en un espacio de características de mayor dimensión.

- Random Forest Classifier:

Es un ensamble de muchos árboles de decisión, cada uno entrenado con una muestra diferente del conjunto de datos (bagging). La predicción se realiza por votación mayoritaria.

Algoritmo:

1). Para cada árbol toma una muestra aleatoria con reemplazo (bootstrap) del dataset. Construye un árbol de decisión sin poda. En cada división del árbol, se selecciona un subconjunto aleatorio de características para decidir la mejor partición.

2). Para predecir se pasa la entrada por todos los árboles. Cada uno da una predicción. Se elige la clase por votación mayoritaria.

No tiene una única función objetivo explícita, pero cada árbol minimiza una medida de impureza como:

Gini: $G = 1 - \sum_k P_k^2$

Entropía: $H = - \sum_k P_k \log P_k$

P: Número de características del dataset (dimensión del vector de entrada).

K: Número de clases (output categories).

- Gaussian Process classifier (GPC):

Es un enfoque no paramétrico y probabilístico. Define una distribución sobre funciones y utiliza un proceso Gaussiano como prior sobre esas funciones.

Para clasificación:

Se define una función latente $f(x) \sim \mathcal{GP}(\mu, K(x, x'))$

Luego se pasa por una función sigmoide para obtener una probabilidad:

$$P(y=1|x) = \sigma(f(x)) = \frac{1}{1 + e^{-f(x)}}$$

Problema de optimización:

Como el likelihood no es gaussiano, se usa una aproximación (ej: Laplace o Expectation Propagation).

$$\max_F \log P(y|F) - \frac{1}{2} F^T K^{-1} F$$

K: matriz de covarianza construida con el kernel $K(x_i, x_j)$.

F: valores de la función latente por cada dato.

y: etiquetas observadas

La ventaja que tiene GPC es que da incertidumbre sobre las predicciones.

- clasificador basado en Deep Learning (ej: CNN)

Una Red Neuronal convolucional (CNN) es ideal para imágenes. Aprende Filtros convolucionales que capturan patrones especiales (bordes, texturas, formas).

Estructura típica:

- Capas convolucionales: aplican Filtros que se aprenden
- Funciones de activación: ReLU.
- Capas de pooling: reducción de dimensionalidad.
- Capas Fully connected: para clasificación

Función objetivo: (clasificación multi-clase).

$$\min_{\theta} L(y, \hat{y}) = - \sum_{i=1}^n \sum_{k=1}^K y_{ik} \log(\hat{y}_{ik})$$

\hat{y}_{ik} : salida softmax de la red para clase k

y_{ik} : etiqueta one-hot.

Se entrena con:

- Backpropagation
- Optimizadores como SGD, Adam.
- Regularización: Dropout, batch normalization, etc.