Support Vector Machine

Minería de Datos (CC3074) - 2026

Support Vector Machine

Semestre 01, 2026

Problematica

Los modelos de clasificación buscan una frontera entre clases.

Pero hay infinitas fronteras posibles.

¿Cuál es la mejor?

Intuición

Imagina dos grupos de puntos en un plano.

Se puede trazar infinitas líneas que los separen.

Algunas líneas pasan muy cerca de los puntos.

Una pequeña perturbación bastaría para clasificar mal una instancia nueva.

La pregunta clave

No solo queremos separar.

Queremos separar con la mayor seguridad posible.

Queremos que la frontera esté lo más alejada de ambas clases.

Definición

SVM es un algoritmo de aprendizaje supervisado.

Se utiliza principalmente para clasificación.

Su objetivo es encontrar el hiperplano que maximiza el margen entre las clases.

Hiperplano

En 2 dimensiones: una línea.

En 3 dimensiones: un plano.

En N dimensiones: un hiperplano de N−1 dimensiones.

Es la frontera de decisión de SVM.

Ecuación

`$w · x + b = 0$`

Donde:

- w → vector normal al hiperplano
- x → instancia de entrada
- b → sesgo (desplazamiento)

Una nueva instancia se clasifica según el signo de `$w · x + b$`

El margen

El margen es la distancia entre la frontera y los puntos más cercanos de cada clase.

Frontera y márgenes

SVM define tres líneas paralelas:

- `$w · x + b = 0$` → la frontera de decisión
- `$w · x + b = +1$` → borde del margen positivo
- `$w · x + b = -1$` → borde del margen negativo

El margen total es la distancia entre las dos líneas exteriores.

Maximizar el margen

`$margen = 2 / ||w||$`

Maximizar el margen equivale a minimizar `$||w||$`.

Mayor margen → mayor capacidad de generalización.

El modelo es menos sensible a perturbaciones en los datos.

Vectores de soporte

Los vectores de soporte son los puntos de entrenamiento más cercanos al hiperplano.

Son exactamente los que definen la posición de la frontera.

Importancia

Si se eliminan los vectores de soporte, la frontera cambia.

Si se eliminan otros puntos, la frontera no cambia.

Solo un subconjunto de los datos define el modelo.

Consecuencia práctica

SVM es eficiente en memoria:

- Solo necesita los vectores de soporte para hacer predicciones.
- No necesita todos los datos de entrenamiento en producción.

Margen duro

El margen duro exige que todos los puntos estén correctamente clasificados.

No se permite ningún error.

Limitaciones

- Solo funciona con datos perfectamente separables.
- Extremadamente sensible a outliers.
- Un solo punto mal ubicado puede hacer imposible la solución.

Ejemplo

Un cliente con características atípicas puede forzar una frontera muy diferente.

El modelo sacrifica generalización para satisfacer un solo punto extremo.

Margen blando

En la práctica, los datos no son perfectamente separables.

El margen blando permite cierta tolerancia a errores.

Parámetro C

El parámetro C controla la penalización por errores.

| C | Comportamiento | | --- | --- | | C alto | Poca tolerancia → margen estrecho → posible sobreajuste | | C bajo | Más tolerancia → margen amplio → mejor generalización | | C = 1 | Valor por defecto, punto de partida razonable |

Trade-off

C alto → el modelo intenta clasificar bien cada punto de entrenamiento.

C bajo → el modelo prefiere un margen más amplio aunque cometa algunos errores.

Es el mismo trade-off varianza / sesgo de siempre.

El truco del kernel

¿Qué pasa cuando los datos no son separables con una línea recta?

Datos no lineales

Muchos problemas reales no tienen clases separables linealmente.

Una frontera recta no es suficiente.

La idea

Si los datos no son separables en 2D, quizás lo sean en 3D.

Se transforma el espacio de características a una dimensión mayor.

En ese nuevo espacio, se puede trazar un hiperplano lineal.

Al proyectarlo de vuelta, la frontera parece curva.

El truco

Transformar explícitamente los datos a alta dimensión es costoso.

Los kernels calculan el producto punto en ese espacio de forma implícita.

No hace falta calcular las coordenadas transformadas.

Misma potencia, menor costo computacional.

Tipos de kernel

Kernel lineal

`$K(x, x') = x · x'$`

No transforma el espacio.

Equivale al SVM clásico.

Funciona bien cuando los datos son linealmente separables o de alta dimensión.

Ideal para: clasificación de texto, datos con muchas variables.

Kernel polinomial

`$K(x, x') = (γ x · x' + r)^d$`

Crea fronteras de decisión curvas.

El parámetro `$d$` controla el grado del polinomio.

Kernel RBF (Gaussiano)

`$K(x, x') = exp(-γ ||x - x'||²)$`

Es el kernel más usado en la práctica.

Mide similitud por proximidad: puntos cercanos → alta similitud.

Crea fronteras muy flexibles que envuelven los grupos.

Elección del kernel

| Kernel | Cuándo usarlo | | --- | --- | | Lineal | Datos de texto, alta dimensionalidad, muchas variables | | Polinomial | Datos de imagen, relaciones de orden superior | | RBF | Casi todo lo demás; punto de partida recomendado |

El parámetro gamma

El parámetro gamma controla el alcance de cada punto de entrenamiento.

Solo aplica para kernels RBF, polinomial y sigmoide.

Efecto de gamma

| Gamma | Efecto | | --- | --- | | Gamma bajo | Alcance amplio → frontera suave, global | | Gamma alto | Alcance estrecho → frontera ajustada, local |

Interacción con C

Gamma alto + C alto → sobreajuste severo.

Ambos hiperparámetros deben ajustarse juntos, no de forma independiente.

Comparación con otros modelos

| Aspecto | SVM | Árbol | Reg. Logística | KNN | | --- | --- | --- | --- | --- | | Frontera de decisión | Lineal o no lineal | No lineal | Lineal | No lineal | | Interpretabilidad | Baja | Alta | Alta | Baja | | Alta dimensionalidad | Excelente | Deficiente | Buena | Deficiente | | Velocidad de entrenamiento | Lenta | Rápida | Rápida | Sin entrenamiento | | Requiere escalado | Sí | No | Sí | Sí | | Clases desbalanceadas | Necesita ajuste | Necesita ajuste | Necesita ajuste | Necesita ajuste |

¿Cuándo usar SVM?

- Pocos datos (menos de ~100,000 instancias)
- Alta dimensionalidad (más variables que instancias)
- Margen de separación razonablemente claro
- Clasificación binaria

¿Cuándo no usar SVM?

- Datasets con millones de filas
- Se necesitan probabilidades directas
- Se requiere interpretabilidad del modelo
- Los datos son muy ruidosos con gran solapamiento de clases

Aplicaciones en minería de datos

Detección de spam

Cada correo se convierte en un vector de miles de palabras.

SVM con kernel lineal encuentra la frontera óptima entre spam y no spam.

Era el estándar en filtros de correo antes del aprendizaje profundo.

Diagnóstico médico

El dataset de cáncer de mama (569 muestras, 30 características) es un caso clásico.

SVM clasifica tumores como malignos o benignos.

Exactitud típica: 95 – 97%.

Detección de fraude financiero

Las transacciones fraudulentas tienen patrones distintos al comportamiento normal.

SVM encuentra la frontera que separa transacciones legítimas de sospechosas.

Se ajusta el parámetro `class_weight='balanced'` para clases desbalanceadas.

Clasificación de imágenes

Antes de las redes neuronales profundas, SVM con kernel RBF era el estado del arte.

Aún aparece en pipelines de imagen médica donde los datos son escasos.

Hiperparámetros

C — regularización

Penalización por errores de clasificación.

Rango típico a explorar: `[0.001, 0.01, 0.1, 1, 10, 100]`

gamma — alcance del kernel

Controla cuánto influye cada punto de entrenamiento.

Rango típico: `['scale', 'auto', 0.001, 0.01, 0.1, 1]`

`scale` es el valor por defecto y un buen punto de partida.

kernel

El tipo de transformación del espacio.

Probar siempre en este orden:

1. Lineal (más rápido)
2. RBF (más flexible)
3. Polinomial (casos específicos)

Búsqueda de hiperparámetros

Los tres parámetros interactúan entre sí.

No se optimizan uno a la vez.

Se usa búsqueda en grilla con validación cruzada para encontrar la combinación óptima.

Ventajas

- Efectivo en espacios de alta dimensión.
- Eficiente en memoria: solo almacena los vectores de soporte.
- Versátil: el kernel define el tipo de frontera.

- Robusto contra overfitting en espacios de alta dimensión.
- Funciona bien cuando hay pocos datos de entrenamiento.

Limitaciones

- No escala bien con datasets grandes (complejidad cuadrática o cúbica).
- Sensible a la escala de las variables: normalización obligatoria.
- La selección del kernel y los hiperparámetros requiere experiencia y tiempo.
- Difícil de interpretar: no explica fácilmente por qué tomó una decisión.
- No produce probabilidades directamente.

La maldición de la dimensionalidad invertida

A diferencia de KNN, SVM se beneficia de la alta dimensionalidad.

Más dimensiones → mayor probabilidad de que los datos sean separables.

Es una de las razones por las que funciona tan bien con texto.

Escalado obligatorio

SVM mide distancias entre puntos.

Si una variable va de 0 a 1 y otra de 0 a 100,000:

La segunda domina completamente el cálculo.

Consecuencia

El modelo ignora las variables de escala pequeña.

La frontera se construye basada casi exclusivamente en la variable grande.

El modelo es inútil sin normalización.

Solución

Aplicar StandardScaler o MinMaxScaler antes de entrenar.

En scikit-learn, usar Pipeline para evitar fuga de datos.

¿Qué puede salir mal?

- No escalar los datos → el modelo ignora variables de escala pequeña
- C muy alto → sobreajuste: funciona en entrenamiento, falla en datos nuevos
- C muy bajo → subajuste: la frontera es demasiado simple para el problema
- Gamma alto con kernel RBF → frontera excesivamente ajustada a los puntos
- Clases desbalanceadas sin `class_weight='balanced'` → sesgo hacia la clase mayoritaria
- Ajustar C y gamma por separado en lugar de juntos → resultados engañosos