
Clase 065 — Precision/Recall tradeoff

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 3. Duración estimada: 50 min.

Clase 065 — Precision/Recall tradeoff

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 3. Duración estimada: 50 min.

Objetivo

Que el alumno entienda que no se puede maximizar precision y recall al mismo tiempo: mover el threshold de decisión sube uno y baja el otro. La clase enseña a usar `decision_function` + `precision_recall_curve` para elegir el threshold según el costo del negocio, no según el default de 0.

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Explicar el tradeoff entre precision y recall en términos del threshold del clasificador.
2. Obtener scores crudos con `decision_function(X)` (o `predict_proba`) en vez de quedarse con `predict`.
3. Calcular la curva con `precision_recall_curve(y_true, scores)` y graficarla.
4. Elegir un threshold que cumpla una restricción del negocio (ej: $\text{precision} \geq 90\%$).
5. Reportar `average_precision_score` como métrica resumen única de la curva.

Temas

#	Tema	Por qué importa
1	Threshold de decisión: el default 0 no es	Define cuántos FP y FN tolerás.
2	<code>decision_function</code> vs <code>predict_proba</code> vs <code>pred</code>	El primero devuelve score crudo, el último
3	<code>precision_recall_curve</code>	Te da los 3 arrays: precision, recall, thr
4	Elegir threshold según restricción de nego	"Precision $\geq 90\%$ " o "Recall $\geq 95\%$ " — depen
5	<code>average_precision_score</code> (AP)	Resumen escalar del área bajo la curva PR.
6	Cuándo PR > ROC	Datasets muy desbalanceados (la próxima cl

Definiciones y características

`decision_function(X)`

: Devuelve el score crudo del clasificador (distancia al hiperplano en SGD/SVM). Valores > threshold → clase positiva. El default de `predict` es `threshold = 0`.

Threshold de decisión

: Punto de corte sobre el score. Subirlo → menos positivos predichos → más precision, menos recall. Bajarlo → más positivos predichos → más recall, menos precision.

`precision_recall_curve(y_true, scores)`

: Devuelve (precisions, recalls, thresholds) para todos los thresholds posibles. Notar que precisions y recalls tienen un elemento más que thresholds (el extremo donde `recall=0`).

`precision_score(y_true, y_pred)`

: $\text{TP} / (\text{TP} + \text{FP})$. "De los que predije positivos, cuántos lo eran". Costo de FP alto → maximizar precision.

`recall_score(y_true, y_pred)`

: $TP / (TP + FN)$. "De los positivos reales, cuántos encontré". Costo de FN alto (cáncer, fraude) → maximizar recall.

`average_precision_score(y_true, scores)` (AP)

: Área bajo la curva precision-recall, calculada como promedio ponderado de precisions en cada threshold. Métrica resumen — mejor que F1 cuando hay desbalance fuerte.

`cross_val_predict(..., method='decision_function')`

: Variante de `cross_val_predict` que devuelve scores crudos en vez de labels. Imprescindible para construir la curva PR sin leakage.

Dataset / recursos

MNIST (clasificador binario "es el 5 vs no") — mismo dataset que la clase 056. Permite reusar el `SGDClassifier` ya entrenado.

Ejercicios

- Score crudo. Entrená `SGDClassifier` sobre MNIST binario "es 5". Para una imagen concreta, llamá `sgd.decision_function([X[0]])` y compará con `sgd.predict([X[0]])`. Mostrá que `predict` es `decision_function > 0`.
- Curva PR. Con `cross_val_predict(sgd, X_train, y_train_5, cv=3, method='decision_function')` obtené `y_scores`. Pasalos a `precision_recall_curve` y graficá precision y recall vs threshold en el mismo eje.
- Threshold para precision $\geq 90\%$. Encontrá el threshold mínimo que garantice `precision >= 0.90`. Pista: `thresholds[np.argmax(precisions >= 0.90)]`. Aplícalo: `y_pred_90 = (y_scores >= threshold_90)` y verificá precision y recall resultantes.
- Curva precision vs recall. Graficá precision (eje Y) contra recall (eje X) — la forma canónica de la curva PR. Marcá el punto correspondiente al threshold por default (0).
- Average precision. Calculá `average_precision_score(y_train_5, y_scores)`. Compará con el F1 que sacaste en la clase 056. ¿Cuál te parece más informativo?

Homework verificable

Notebook con MNIST binario (es 5 vs no): (a) entrenar `SGDClassifier`; (b) obtener `y_scores` con `cross_val_predict(method='decision_function')`; (c) graficar las dos curvas (precision/recall vs threshold y precision vs recall); (d) encontrar el threshold que da precision $\geq 90\%$ y reportar el recall en ese punto; (e) reportar `average_precision_score`.

Criterio de aceptación: El threshold elegido para precision $\geq 90\%$ verifica esa cota cuando se aplica sobre `y_scores`. Las dos curvas están graficadas con ejes etiquetados.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
precisions y thresholds tienen distinto la	Es por diseño: <code>precision_recall_curve</code> devu

Usás predict y querés mover el threshold	predict ya colapsó el score a label. Fix:
Sacaste la curva con y_pred en vez de y_sc	La curva PR necesita scores continuos, no
Threshold elegido sobre train, no sobre va	Overfitting del threshold. Fix: elegí thre
Modelos sin decision_function (ej: RandomForest)	No todos los estimadores la exponen. Fix:

Preguntas frecuentes

¿Precision o recall — cuál priorizo?

Depende del costo asimétrico. Filtro de spam: un FP (mail bueno marcado como spam) cuesta más que un FN → priorizá precision. Detección de cáncer: un FN (cáncer no detectado) cuesta vidas → priorizá recall. No hay respuesta universal, hay análisis de costos.

¿decision_function o predict_proba?

Son intercambiables a los fines de la curva PR (ambas son scores monótonos). predict_proba devuelve probabilidades calibradas en [0,1] (más interpretable); decision_function devuelve scores sin acotar. Para SGDClassifier usá decision_function; para RandomForestClassifier usá predict_proba(X)[:, 1].

¿Por qué la curva PR tiene esa forma escalonada?

Cada salto corresponde a un sample que cambia de lado del threshold. Con N muestras hay hasta N thresholds distintos. En datasets chicos se nota más; con 10k+ samples se ve suave.

¿F1 o average precision?

F1 es a un threshold fijo (típicamente 0.5) — útil cuando el threshold ya está definido. Average precision integra sobre todos los thresholds — útil para comparar modelos sin fijar threshold. Para selección de modelo, AP es más informativo.

¿Y si la clase positiva es la minoría extrema (<1%)?

Ahí la curva PR brilla y la ROC engaña. Lo cierra la clase 058 (ROC y AUC).

Referencias

- Géron, cap. 3 § Precision/Recall Tradeoff y § The ROC Curve.
- scikit-learn — precision_recall_curve
- scikit-learn — average_precision_score
- scikit-learn — Precision-Recall example

Siguiente clase

Clase 066 — Curva ROC y AUC

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb