Clase 091 — La maldición de la dimensionalidad

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 8 § The Curse of Dimensionality. · Duración estimada: 45 min.

Clase 091 — La maldición de la dimensionalidad

> *Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 8 § The Curse of Dimensionality. · Duración estimada: 45 min.*

Objetivo

Que el alumno entienda por qué los algoritmos basados en distancia (kNN, k-means, SVM-RBF) degradan en alta dimensión: el espacio se vuelve mayormente vacío, las distancias entre puntos colapsan a un mismo valor, y los modelos overfittean. Esto motiva la reducción de dimensionalidad (PCA, manifold learning) que viene en las próximas clases.

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Explicar la sparsity exponencial: por qué llenar uniformemente un hipercubo unitario requiere n^d puntos.
2. Calcular numéricamente que en d=100 la razón (d_max - d_min) / d_min entre distancias euclidianas tiende a 0.
3. Identificar qué algoritmos sufren la maldición (basados en distancia/densidad) y cuáles menos (árboles, modelos lineales con regularización).
4. Reconocer la manifold hypothesis como justificación de PCA, t-SNE, UMAP: los datos reales viven en un subespacio de baja dimensión.
5. Decidir cuándo reducir dimensionalidad vs. cuándo regularizar o usar otro modelo.

Temas

#	Tema	Por qué importa
1	Intuición geométrica: hipercubo y volumen	En d=100, el 99.99% del volumen está pegad
2	Sparsity exponencial	Para cubrir el espacio uniformemente, los
3	Concentración de la medida	Distancias entre puntos aleatorios converg
4	Distancia euclidiana pierde sentido	nearest neighbor deja de ser informativo.
5	Hubness	Algunos puntos aparecen como vecinos de to
6	Manifold hypothesis	Los datos reales no llenan el espacio: viv
7	Implicaciones prácticas para ML	Overfitting, kNN colapsa, k-means inestabl

Definiciones y características

Maldición de la dimensionalidad

: Conjunto de fenómenos contraintuitivos que aparecen al crecer la cantidad de features: el espacio se vuelve mayormente vacío, las distancias se igualan, y la cantidad de datos necesaria para densidad constante crece exponencialmente en d. Acuñada por Bellman (1961).

Sparsity exponencial

: Para mantener una densidad constante de puntos en un hipercubo [0,1]^d, hace falta n^d muestras. Con d=100 y n=10 por eje son 10^100 puntos — más que átomos en el universo observable. Consecuencia: en alta dimensión todos los datasets son chicos.

Concentración de la medida

: En distribuciones de alta dimensión (uniforme, gaussiana), la distancia entre dos puntos aleatorios se concentra fuertemente alrededor de su media. Formalmente, Var(||X-Y||) / E[||X-Y||]^2 → 0 cuando d → ∞. Resultado: d_min ≈ d_max, y la noción de "más cercano" se vuelve ruido.

Distancia euclidiana en alta dimensión

: Pierde poder discriminativo porque acumula varianza por cada feature irrelevante. Alternativas: distancia coseno (escala-invariante), Mahalanobis (decorrela), o reducir antes con PCA.

Hubness

: Fenómeno por el cual, en alta dimensión, ciertos puntos aparecen entre los k vecinos más cercanos de muchísimos otros (hubs), mientras otros nunca son vecinos de nadie (anti-hubs). Rompe el supuesto de simetría que asume kNN.

Manifold hypothesis

: Suposición de que los datos reales de alta dimensión (imágenes, texto, audio) no llenan el espacio ambiente sino que viven en un subespacio (manifold) de dimensión intrínseca mucho menor. Justifica PCA, autoencoders, t-SNE, UMAP. Sin esta hipótesis, reducir dimensionalidad sería destruir información.

Dimensión intrínseca

: Cantidad mínima de variables latentes necesarias para representar los datos sin pérdida apreciable. MNIST tiene d=784 features (28×28 pixels) pero dimensión intrínseca estimada ~10–15.

Dataset / recursos

Sintético: puntos uniformes en [0,1]^d para d {2, 10, 100, 1000}. Permite calcular numéricamente sparsity, ratio d_max/d_min, y fracción de volumen cerca del borde. Para la parte aplicada, sklearn.datasets.load_digits (d=64) como ejemplo de manifold hypothesis empírica.

Ejercicios

1. Volumen del borde. Calculá la fracción de volumen del hipercubo [0,1]^d que está a menos de 0.01 del borde, para d=2, 10, 100. Fórmula: 1 - 0.98^d.

2. Concentración de distancias. Sampleá 1000 puntos uniformes en [0,1]^d. Para d {2, 10, 100, 1000}, calculá (d_max - d_min) / d_min sobre todas las distancias pairwise. Verificá que tiende a 0.

3. Distancia al vecino más cercano. Con n=1000 puntos uniformes y d variable, graficá la distancia media al 1-NN. Mostrá que crece con d (el "vecino" está cada vez más lejos).

4. kNN degrada. Generá clasificación sintética con n=500, agregando d features de ruido puro (irrelevantes). Evaluá accuracy de KNeighborsClassifier para d = 2, 10, 50, 200. Curva debería caer.

5. Manifold hipótesis empírica. Cargá sklearn.datasets.load_digits. Calculá cuántos componentes PCA explican el 95% de la varianza vs. d=64 originales — estimación grosera de dimensión intrínseca.

Homework verificable

Notebook que: (a) genere puntos uniformes en [0,1]^d para d [1, 200]; (b) calcule y grafique (d_max - d_min)/d_min vs d; (c) entrene KNeighborsClassifier con n=1000 y features añadidas de ruido N(0,1), reporte accuracy vs d; (d) sobre load_digits, calcule cuántos componentes PCA explican 90%, 95%, 99% de varianza.

Criterio de aceptación: El ratio de distancias tiende a 0 para d > 50. Accuracy de kNN cae monotónicamente al agregar ruido. PCA muestra que digits (d=64) tiene dimensión intrínseca ≤ 30 al 95%.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
"Agregué más features y el modelo empeoró"	Features irrelevantes inyectan ruido y emp
kNN con k=1 da resultados aleatorios en al	Concentración de medida: d_min ≈ d_max, "e
"PCA me bajó accuracy"	Te quedaste con muy pocos componentes, o e
Confundir d (features) con n (muestras)	"Tengo 10M filas, no aplica la maldición"
Pensar que árboles también sufren	Random Forest y boosting son mucho más rob

Preguntas frecuentes

¿A partir de qué d empiezo a preocuparme?

Heurística: con kNN/k-means/SVM-RBF, ya con d > 20–30 y n modesto se nota la degradación. Con árboles y modelos lineales regularizados, podés escalar a d = 10000 sin drama (texto TF-IDF, genómica).

¿Más datos resuelve la maldición?

Solo asintóticamente, y la cantidad necesaria crece exponencialmente. Para d=100 no hay suficientes datos en el universo. La salida real es reducir d (PCA, feature selection) o usar modelos que no dependan de densidad global.

¿Por qué deep learning funciona con d enorme (imágenes 224×224×3 = 150k)?

Por la manifold hypothesis: las imágenes naturales no llenan R^150000, viven en un manifold de dimensión intrínseca mucho menor. Las redes profundas aprenden esa estructura jerárquicamente. La maldición sigue aplicando si hacés kNN sobre pixels crudos — y por eso no se hace.

¿Distancia coseno me salva?

Mitiga, no salva. Es invariante a escala y funciona mejor en texto (TF-IDF, embeddings), pero también sufre concentración si las features son ruido puro. Mejor combinar coseno + reducción + selección.

¿Cómo estimo la dimensión intrínseca?

Métodos: (a) PCA + curva de varianza explicada (rápido, lineal); (b) sklearn.manifold (Isomap, LLE); (c) estimadores específicos como MLE de Levina–Bickel o el paquete skdim. Para empezar, PCA al 95% alcanza como aproximación grosera.

Referencias

- Géron, cap. 8 § The Curse of Dimensionality y § Main Approaches for Dimensionality Reduction.
- Bellman, R. (1961). Adaptive Control Processes: A Guided Tour — origen del término.

- Beyer, K. et al. (1999). When Is "Nearest Neighbor" Meaningful? — paper clásico sobre concentración de distancias.
- Radovanović, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data — fenómeno de hubness.
- scikit-learn — Manifold learning

Siguiente clase

Clase 092 — PCA: proyección, varianza explicada, incremental, randomized, kernel

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb