Clase 093 — LLE (Locally Linear Embedding)

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 8. Duración estimada: 45 min.

Clase 093 — LLE (Locally Linear Embedding)

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 8. Duración estimada: 45 min.

Objetivo

Entender LLE como técnica no lineal de reducción de dimensionalidad basada en manifold learning: preservar las relaciones lineales locales entre cada punto y sus vecinos para "desenrollar" estructuras curvas (Swiss roll, S-curve) donde PCA falla.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Explicar la intuición de LLE: cada punto se reconstruye como combinación lineal de sus k vecinos, y esa relación se preserva en baja dimensión.
- Aplicar sklearn.manifold.LocallyLinearEmbedding sobre un dataset no lineal (Swiss roll) y visualizar el resultado en 2D.
- Elegir el hiperparámetro n_neighbors y discutir su impacto (sub/sobre-ajuste local).
- Comparar LLE contra PCA y otras técnicas no lineales (Isomap, t-SNE) en términos de qué preservan.
- Identificar cuándo LLE es apropiado y cuándo conviene usar la variante Modified LLE.

Temas

- Motivación: limitaciones de PCA en manifolds curvos.
- Algoritmo LLE en dos pasos: (1) pesos de reconstrucción local, (2) embedding que preserva esos pesos.
- Hiperparámetros: n_neighbors, n_components, method (standard, modified, hessian, ltsa).
- Variantes: Modified LLE (MLLE), Hessian LLE, LTSA.
- Visualización del Swiss roll antes y después.
- Limitaciones: costo computacional $O(m \log(m) \cdot n \cdot k^3 + m \cdot n \cdot k^2)$ y sensibilidad al ruido.

Definiciones y características

- LLE (Locally Linear Embedding): técnica no lineal de reducción de dimensionalidad. No usa proyecciones; descubre cómo cada instancia se relaciona linealmente con sus vecinos más cercanos y busca una representación de baja dimensión donde esas relaciones locales se preserven al máximo.
- Manifold learning: familia de técnicas que asume que los datos de alta dimensión yacen sobre una variedad (manifold) de menor dimensión embebida en el espacio original. LLE, Isomap, t-SNE y UMAP son ejemplos.
- k vecinos (n_neighbors): cantidad de vecinos más cercanos que se usan para reconstruir cada punto. Valor bajo → ruido / desconexiones; valor alto → pierde la estructura local y se parece a PCA.
- Reconstruction weights (W): matriz de pesos que minimiza $\Sigma \, ||x - \Sigma_j \, w_{ij} \, x^j||^2$ con $\Sigma_j \, w_{ij} = 1$ y $w_{ij} = 0$ si j no es vecino de i. Captura la geometría local.
- Modified LLE (MLLE): variante que usa múltiples vectores de pesos por vecindario para evitar el problema de regularización del LLE estándar cuando n_neighbors > n_components. Más estable y

suele dar mejores embeddings.
- Hessian LLE / LTSA: otras variantes que reemplazan el criterio de reconstrucción por restricciones geométricas más fuertes (curvatura local, tangentes locales).

Dataset / recursos

- sklearn.datasets.make_swiss_roll(n_samples=1000, noise=0.2) — dataset clásico para visualizar reducción no lineal.
- sklearn.datasets.make_s_curve(n_samples=1000) — manifold alternativo en forma de S.
- sklearn.manifold.LocallyLinearEmbedding.
- Géron, cap. 8 — sección "LLE" y figuras del Swiss roll.

Ejercicios

1. Swiss roll básico: generá un Swiss roll de 1000 puntos, aplicá LocallyLinearEmbedding(n_neighbors=10, n_components=2) y graficá el resultado coloreando por la coordenada original t. Verificá que el rollo quede "desenrollado".
2. Comparación con PCA: sobre el mismo Swiss roll, aplicá PCA(n_components=2) y compará visualmente. Discutí por qué PCA aplasta el rollo en lugar de desenrollarlo.
3. Barrido de n_neighbors: probá n_neighbors ∈ {5, 10, 30, 100} y graficá los 4 embeddings en una grilla 2×2. Describí qué pasa en cada extremo.
4. Modified LLE: repetí el ejercicio 1 con method='modified' y n_neighbors=12. Compará con el LLE estándar — ¿qué embedding se ve más limpio?
5. LLE sobre datos reales: cargá load_digits() (64 dimensiones) y proyectá a 2D con LLE. Coloreá por dígito. ¿Se separan las clases?

Homework verificable

Generá un Swiss roll con n_samples=1500 y random_state=42. Aplicá LLE estándar y MLLE (ambos con n_neighbors=12, n_components=2). Guardá los dos arrays resultantes en embeddings.npz con keys lle y mlle.

Criterio de aceptación: ambos arrays tienen shape (1500, 2), ningún NaN, y la correlación de Spearman entre la primera coordenada del embedding de MLLE y la variable t del Swiss roll original es $|\rho| > 0.95$ (el embedding preservó el orden a lo largo del rollo).

Errores comunes

- No escalar los datos cuando las features tienen magnitudes muy distintas — los "vecinos" pasan a estar dominados por una feature.
- Elegir n_neighbors muy bajo (ej. 3-4): el grafo de vecindad queda desconectado y LLE devuelve embeddings rotos o con componentes colapsados.
- Elegir n_neighbors muy alto: se pierde el carácter local y el resultado tiende a parecerse a PCA, perdiendo la ventaja no lineal.
- Esperar que LLE preserve distancias globales: no lo hace. Para distancias geodésicas usar Isomap; para clusters bien separados, t-SNE o UMAP.
- Usar LLE estándar con n_neighbors > n_components sin regularización: la solución se vuelve degenerada. Para eso existe MLLE.

Preguntas frecuentes

- ¿PCA o LLE? PCA si los datos viven en un subespacio lineal o si querés interpretar componentes / hacer compresión rápida. LLE (y otras técnicas de manifold) si la estructura es curva y querés visualizar. En la práctica, primero PCA para reducir ruido a ~50 dimensiones y después LLE/t-SNE a 2D.
- ¿LLE sirve para preprocesar antes de un clasificador? Rara vez. No es eficiente sobre datos nuevos (no tiene una función transform natural y rápida) y t-SNE/UMAP suelen visualizar mejor. Para preprocesar, PCA o autoencoders.
- ¿Qué diferencia hay con Isomap? Isomap preserva distancias geodésicas globales sobre el grafo de vecinos; LLE preserva sólo relaciones lineales locales. Isomap suele dar embeddings más fieles a la geometría global; LLE es más barato.
- ¿Por qué a veces LLE colapsa todo en una línea? Suele ser un n_neighbors mal elegido o falta de regularización. Probá MLLE o ajustá reg (parámetro de regularización del solver).
- ¿Es determinístico? No del todo: depende del solver de eigendecomposición y de random_state. Fijá random_state y eigen_solver='dense' para reproducibilidad estricta.

Referencias

- Géron, A. Hands-On Machine Learning, 3ra ed., cap. 8 — Dimensionality Reduction, sección "LLE".
- Roweis, S. & Saul, L. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science, 290(5500).
- scikit-learn docs: LocallyLinearEmbedding y manifold learning comparison.
- Zhang, Z. & Wang, J. (2007). MLLE: Modified Locally Linear Embedding Using Multiple Weights.

Siguiente clase

Clase 094 — MDS, Isomap, t-SNE, UMAP, LDA

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb