Una introducción a Local Lineal Embeddings

Contenidos

1 Introducción al Problema de Reducción de Dimensionalidad

1.1 Contexto y la Maldición de la Dimensionalidad

En el **Aprendizaje Estadístico/ Automático** , trabajamos con conjuntos de datos $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, donde cada observación $\mathbf{x}_i$ reside en un espacio de alta dimensión, $\mathbf{x}_i \in \mathbb{R}^D$. El número de dimensiones D puede ser muy grande. Esta alta dimensionalidad introduce serios desafíos computacionales y estadísticos, a menudo resumidos como la **Maldición de la Dimensionalidad** .

Cuando D es excesivamente grande: * El volumen del espacio $\mathbb{R}^D$ crece exponencialmente, lo que implica que la nocion de cercania se pierda. * Los algoritmos pueden sufrir de **sobreajuste**, capturando ruido en lugar de la estructura subyacente debido a este problema.

1.2 Formulación Matemática de la Reducción de Dimensionalidad

El objetivo de la **Reducción de Dimensionalidad** es encontrar una representación de baja dimensión de los datos, $\mathbf{y}_i \in \mathbb{R}^d$, tal que $d \ll D$, preservando tanto la estructura local y global del conjunto de datos.

Formalmente, buscamos una función de mapeo $f : \mathbb{R}^D \to \mathbb{R}^d$ que transforma la matriz de datos de alta dimensión $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ en una matriz de baja dimensión $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N] \in \mathbb{R}^{d \times N}$:

$$\mathbf{y}_i = f(\mathbf{x}_i), \quad \forall i \in \{1, \dots, N\}$$

1.2.1 La Hipótesis de la Variedad (Manifold Hypothesis)

Una suposición fundamental en muchos métodos no lineales, como *Local Linear Embedding* (LLE), es la **Hipótesis de la Variedad**. Postulamos que, aunque los datos se observan en $\mathbb{R}^D$, en realidad se encuentran cercanos a una variedad (subespacio) embebida $\mathcal{M}$ de dimensión d (donde $d \ll D$):

$$\mathbf{X} \approx \mathcal{M} \subset \mathbb{R}^D$$

El problema de la reducción de dimensionalidad se transforma entonces en un problema de aprendizaje de variedades (*Manifold Learning*): encontrar el mapeo f que “despliega” o proyecta la variedad $\mathcal{M}$ en el espacio $\mathbb{R}^d$.

1.2.2 Problema de Optimización General

El mapeo óptimo f (o su resultado, $\mathbf{Y}$) generalmente se encuentra minimizando una **función de costo** $\mathcal{L}(\mathbf{Y}, \mathbf{X})$, la cual cuantifica la distorsión o la pérdida de información de la proyección:

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y} \in \mathbb{R}^{d \times N}} \mathcal{L}(\mathbf{Y}, \mathbf{X})$$

Para métodos basados en la estructura local, como LLE, la función $\mathcal{L}$ se diseñará para preservar las **relaciones de vecindad** y las propiedades locales de los datos.

2 Local Linear Embedding (LLE): Idea intuitiva

2.1 Aproximación Lineal Local

Local Linear Embedding (LLE) es un algoritmo de Aprendizaje de Variedades No Lineal (*Non-linear Manifold Learning*) que se basa en dos supuestos geométricos clave:

1. **La Variedad es Localmente Lineal:** Si el *manifold* subyacente $\mathcal{M}$ es suave, cualquier vecindario suficientemente pequeño en $\mathcal{M}$ puede ser aproximado por un subespacio lineal.
2. **Preservación de la Reconstrucción:** Si un punto de datos $\mathbf{x}_i$ puede ser reconstruido como una combinación lineal ponderada de sus vecinos en el espacio de alta dimensión ($\mathbb{R}^D$), esta misma relación de pesos debe ser preservada en el espacio de baja dimensión ($\mathbb{R}^d$).

El algoritmo de LLE se descompone en dos pasos principales de optimización completamente desacoplados.

2.2 Cálculo de los Coeficientes de Reconstrucción (Pesos)

En el primer paso, fijamos los datos $\mathbf{X}$ en $\mathbb{R}^D$ y determinamos los coeficientes de reconstrucción que mejor representan cada punto $\mathbf{x}_i$ como una combinación lineal de sus K vecinos más cercanos, denotados por $\mathcal{N}_i$.

Sea $\mathbf{W} = \{w_{ij}\}_{i,j=1}^{N}$ la matriz de pesos, donde w_{ij} es distinto de cero solo si $\mathbf{x}_j \in \mathcal{N}_i$. El problema de optimización para encontrar $\mathbf{W}$ es:

$$\min_{\mathbf{W}} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{x}_j \right\|^2$$

Sujeto a las siguientes restricciones para garantizar invarianza a traslación y rotación (y para que los pesos definan efectivamente un promedio local):

1. $\sum_{j \in \mathcal{N}_i} w_{ij} = 1$, para todo i.
2. $w_{ij} = 0$, si $\mathbf{x}_j \notin \mathcal{N}_i$.

Importante: La solución a este problema de optimización solo depende de las propiedades locales (tipo de distancia y cantidad de vecinos) de los datos en el espacio de alta dimensión $\mathbb{R}^D$. Una vez calculados, estos pesos w_{ij} **quedan fijos** y son independientes de la dimensión final d.

2.3 Cálculo de las Coordenadas de Baja Dimensión

En el segundo paso, utilizamos la matriz de pesos $\mathbf{W}$ encontrada en el paso 1, y buscamos las coordenadas de baja dimensión $\mathbf{Y} \in \mathbb{R}^{d \times N}$ que mejor conserven esas relaciones de reconstrucción locales.

El objetivo es minimizar el error de reconstrucción en el espacio de baja dimensión, utilizando los pesos $\mathbf{W}$ previamente calculados (**fijos**):

$$\min_{\mathbf{Y}} \sum_{i=1}^{N} \left\| \mathbf{y}_i - \sum_{j=1}^{N} w_{ij} \mathbf{y}_j \right\|^2$$

Sujeto a las siguientes restricciones, necesarias para garantizar una solución no degenerada (es decir, que la solución no colapse todos los puntos a un único punto en el origen):

1. Centrado: $\sum_{i=1}^{N} \mathbf{y}_i = \mathbf{0}$.
2. Normalización: $\frac{1}{N} \sum_{i=1}^{N} \mathbf{y}_i \mathbf{y}_i^T = \mathbf{I}_d$. (La matriz de covarianza de $\mathbf{Y}$ debe ser la matriz identidad $d \times d$).

Este problema se resuelve mediante un problema de valores y vectores propios sobre la matriz Laplaciana de datos $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$.

En las proximas secciones, formalizaremos la solución analítica de estos dos problemas de optimización y exploraremos los resultados del método.

3 Cálculo de las Distancias y Búsqueda de Vecinos

El primer paso de LLE (Sección 3) requiere la identificación de los K vecinos más cercanos ($\mathcal{N}_i$) para cada punto $\mathbf{x}_i$. Este paso es, a menudo, el más costoso computacionalmente.

3.1 Métrica de Distancia Euclidiana

En el contexto de LLE, la cercanía de los puntos en el espacio de alta dimensión $\mathbb{R}^D$ se mide tradicionalmente utilizando la distancia Euclidiana ℓ_2. La distancia entre dos puntos $\mathbf{x}_i$ y $\mathbf{x}_j$ está definida por:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2 = \left(\sum_{m=1}^{D} (x_{im} - x_{jm})^2 \right)^{1/2}$$

Donde x_{im} es la m-ésima componente del vector $\mathbf{x}_i$.

3.2 Búsqueda Clásica de K Vecinos Más Cercanos

El método más directo para encontrar los K vecinos más cercanos de un punto $\mathbf{x}_i$ es la búsqueda exhaustiva. Este procedimiento implica calcular la distancia Euclidiana de $\mathbf{x}_i$ a todos los demás $N - 1$ puntos en el conjunto de datos $\mathbf{X}$ y, posteriormente, seleccionar los K puntos con las distancias mínimas. El costo computacional de este método es el siguiente:

- Cálculo de $N - 1$ distancias Euclidianas por punto.
- Cada cálculo de distancia toma $\mathcal{O}(D)$ tiempo.
- Cálculo total para N puntos

$$\text{Costo Total k-NN (Brute-Force)} = \mathcal{O}(N^2 D)$$

3.3 Aceleración con Árboles de Partición Espacial (Barnes-Hut)

El algoritmo **Barnes-Hut** es fundamental para acelerar la búsqueda de vecinos, un paso clave en métodos como LLE. Logra una reducción en la complejidad, pasando de $\mathcal{O}(N^2)$ a $\mathcal{O}(N \log N \cdot D)$. Esta mejora se debe a que, en lugar de calcular la distancia a cada punto, utiliza una estrategia de partición espacial que aproxima cúmulos distantes de datos mediante sus centroides. En la práctica, esta optimización hace que los algoritmos de reducción de dimensionalidad sean **computacionalmente viables** incluso para conjuntos de datos muy grandes.

$$\text{Costo Total Barnes-Hur} = \mathcal{O}(N \log N \cdot D)$$

Al igual que mucha literatura previa a la popularización de estos métodos, el algorimo LLE vanilla utiliza la búsqueda exhaustiva de vecinos. Sin embargo, en implementaciones actuales se usa barnes-hut.

4 Cálculo de los Coeficientes de Reconstrucción Local

4.1 Formulación del Problema de Mínimos Cuadrados

Una vez que se han identificado los K vecinos más cercanos $\mathcal{N}_i$ para cada punto $\mathbf{x}_i$ (como se detalla en la Sección 4), el siguiente paso del algoritmo Local Linear Embedding (LLE) es determinar los coeficientes de reconstrucción w_{ij}. Estos pesos deben minimizar el error de reconstrucción de $\mathbf{x}_i$ a partir de la combinación lineal de sus vecinos en el espacio de alta dimensión $\mathbb{R}^D$.

El funcional de coste $\mathcal{J}(\mathbf{W})$ para la matriz de pesos $\mathbf{W} \in \mathbb{R}^{N \times N}$ es un problema de Mínimos Cuadrados:

$$\min_{\mathbf{W}} \mathcal{J}(\mathbf{W}) = \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{x}_j \right\|^2$$

Sujeto a las siguientes restricciones para asegurar la invarianza geométrica (traslación y rotación):

1. $\sum_{j \in \mathcal{N}_i} w_{ij} = 1$, para todo $i = 1, \dots, N$.
2. $w_{ij} = 0$, si $\mathbf{x}_j \notin \mathcal{N}_i$.

4.2 Solución Analítica mediante Multiplicadores de Lagrange

El problema de optimización puede resolverse independientemente para cada punto $\mathbf{x}_i$. Sea $\mathbf{w}_i$ el vector de K pesos asociados a los vecinos de $\mathbf{x}_i$. La restricción $\sum w_{ij} = 1$ permite reescribir la expresión de reconstrucción minimizada en términos de los vectores de diferencia $\mathbf{z}_j = \mathbf{x}_j - \mathbf{x}_i$:

$$\min_{\mathbf{w}_i} \mathbf{w}_i^T \mathbf{C}_i \mathbf{w}_i$$

Sujeto a $\mathbf{1}^T \mathbf{w}_i = 1$.

Donde $\mathbf{C}_i \in \mathbb{R}^{K \times K}$ es la **matriz de covarianza local** (o Gramiana local) de las diferencias, definida por el producto interior de los vectores de diferencia entre el punto central y sus vecinos:

$$[\mathbf{C}_i]_{a,b} = \mathbf{z}_a^T \mathbf{z}_b = (\mathbf{x}_a - \mathbf{x}_i)^T (\mathbf{x}_b - \mathbf{x}_i), \quad \mathbf{x}_a, \mathbf{x}_b \in \mathcal{N}_i$$

Utilizando el método de los **Multiplicadores de Lagrange** para incorporar la restricción $\mathbf{1}^T \mathbf{w}_i = 1$, el vector de pesos óptimo $\mathbf{w}_i^*$ es:

$$\mathbf{w}_i^* = \frac{\mathbf{C}_i^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{C}_i^{-1} \mathbf{1}}$$

Si la matriz $\mathbf{C}_i$ es singular (lo que ocurre cuando K es mayor que D o cuando los vecinos son colineales), se debe aplicar una pequeña **regularización** $\delta \mathbf{I}$ a $\mathbf{C}_i$ para asegurar la estabilidad numérica de la inversión: $\mathbf{C}_i \leftarrow \mathbf{C}_i + \delta \mathbf{I}$.

4.3 Algoritmo y Complejidad Computacional del Cálculo de Pesos

El cálculo de los pesos, una vez obtenidos los vecinos, es un proceso iterativo que implica la inversión de N matrices de tamaño $K \times K$.

4.3.1 Algoritmo 1: Cálculo de Pesos LLE

1. **Iteración:** Para cada punto $\mathbf{x}_i$ ($i = 1, \dots, N$), recuperar los K vecinos $\mathcal{N}_i$.
2. **Construcción de $\mathbf{C}_i$:** Formar la matriz de covarianza local $\mathbf{C}_i$ de tamaño $K \times K$.
3. **Inversión:** Calcular la inversa regularizada $\mathbf{C}_i^{-1}$.
4. **Cálculo de $\mathbf{w}_i^*$:** Aplicar la solución analítica para obtener el vector de pesos.
5. **Actualización:** Poblar las entradas correspondientes en la matriz global $\mathbf{W}$.

4.3.2 Análisis de la Complejidad

La complejidad de esta fase no incluye la búsqueda de vecinos (cuya complejidad se detalla en la Sección 3). El costo se centra en la construcción de N matrices Gramianas y sus respectivas inversiones.

Operación	Complejidad por Punto (i)	Complejidad Total
Construcción de $\mathbf{C}_i$	$\mathcal{O}(K^2 D)$	$\mathcal{O}(NK^2 D)$
Inversión de $\mathbf{C}_i$	$\mathcal{O}(K^3)$	$\mathcal{O}(NK^3)$
Total	-	$\mathcal{O}(NK^2 D + NK^3)$

5 Cálculo del Embedding: Solución Numérica mediante SVD

5.1 El Problema de Minimización Cuadrática y Derivación Matricial

El objetivo final de LLE es encontrar la matriz de coordenadas de baja dimensión $\mathbf{Y} \in \mathbb{R}^{d \times N}$ que minimice el error de reconstrucción en el espacio de embedding. Con la matriz de pesos $\mathbf{W}$ (Sección 4) fija, el problema es:

$$\min_{\mathbf{Y}} \mathcal{E}(\mathbf{Y}) = \sum_{i=1}^{N} \left\| \mathbf{y}_i - \sum_{j=1}^{N} w_{ij} \mathbf{y}_j \right\|^2$$

La expresión de arriba se compacta en notación matricial. Sea $\mathbf{A} = \mathbf{I} - \mathbf{W}$, la **matriz de reconstrucción**. El funcional de coste es equivalente a la norma de Frobenius al cuadrado:

$$\mathcal{E}(\mathbf{Y}) = \|\mathbf{Y}(\mathbf{I} - \mathbf{W})\|_F^2 = \|\mathbf{Y}\mathbf{A}\|_F^2$$

Las restricciones esenciales para garantizar una solución única y bien definida son:

1. **Centrado:** $\mathbf{Y}\mathbf{1} = \mathbf{0}$.
2. **Normalización:** $\mathbf{Y}\mathbf{Y}^T = N\mathbf{I}_d$.

5.2 El Mapeo a la Matriz de Mapeo LLE ($\mathbf{M}$)

El funcional de coste se transforma a la forma cuadrática, utilizando la propiedad de la norma de Frobenius $\|\mathbf{B}\|_F^2 = \mathrm{Tr}(\mathbf{B}^T\mathbf{B})$:

$$\mathcal{E}(\mathbf{Y}) = \mathrm{Tr}\left((\mathbf{Y}\mathbf{A})^T(\mathbf{Y}\mathbf{A})\right)$$

Aplicando la transpuesta del producto $(\mathbf{Y}\mathbf{A})^T = \mathbf{A}^T\mathbf{Y}^T$:

$$\mathcal{E}(\mathbf{Y}) = \mathrm{Tr}\left(\mathbf{A}^T\mathbf{Y}^T\mathbf{Y}\mathbf{A}\right)$$

Asumiendo que $\mathbf{Y}$ está definida tal que $\mathbf{Y}^T\mathbf{Y}$ está relacionada con la restricción de normalización (y para mantener la forma canónica del *eigenproblem*):

$$\mathcal{E}(\mathbf{Y}) = \mathrm{Tr}\left(\mathbf{Y}^T\mathbf{M}\mathbf{Y}\right)$$

donde $\mathbf{M} = \mathbf{A}^T\mathbf{A} = (\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$.

El problema se convierte en hallar las d direcciones $\mathbf{y}_k$ (las filas de $\mathbf{Y}$) que minimizan la función objetivo bajo las restricciones ortonormales.

5.3 Solución Mediante Descomposición Espectral

La solución óptima es proporcionada por los **vectores propios** de la matriz $\mathbf{M}$ correspondientes a los **valores propios más pequeños no nulos**.

Para obtener $\mathbf{Y}$, realizamos la descomposición espectral de $\mathbf{M}$:

$$\mathbf{M} = \mathbf{V}\mathbf{V}^T$$

donde es la matriz diagonal de valores propios λ_k, y $\mathbf{V}$ es la matriz de vectores propios $\mathbf{v}_k$.

Los vectores $\mathbf{v}_k$ son ordenados tal que $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N$. * El vector propio $\mathbf{v}_1$ corresponde a $\lambda_1 = 0$ y es descartado. * Las filas (o columnas) de la matriz de embedding $\mathbf{Y}$ se forman con los d vectores propios siguientes: $\mathbf{v}_2, \mathbf{v}_3, ..., \mathbf{v}_{d+1}$.

5.4 Uso de la Descomposición en Valores Singulares (SVD)

Aunque la solución teórica utiliza la descomposición en valores propios de $\mathbf{M}$, en la práctica numérica es más estable y a menudo preferible resolver el problema directamente sobre la matriz central $\mathbf{A} = (\mathbf{I} - \mathbf{W})$ utilizando la **Descomposición en Valores Singulares (SVD)**.

Sea $\mathbf{A} = \mathbf{I} - \mathbf{W}$ la matriz de reconstrucción. La función objetivo es:

$$\min_{\mathbf{Y}} \|\mathbf{Y}\mathbf{A}\|_F^2$$

Aplicando la Descomposición en Valores Singulares a la matriz $\mathbf{A} \in \mathbb{R}^{N \times N}$:

$$\mathbf{A} = \mathbf{U}\mathbf{V}^T$$

Donde: * $\mathbf{U} \in \mathbb{R}^{N \times N}$ y $\mathbf{V} \in \mathbb{R}^{N \times N}$ son matrices ortogonales de vectores singulares. * $\in \mathbb{R}^{N \times N}$ es la matriz diagonal de **valores singulares** σ_k, ordenados de mayor a menor ($\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_N$).

Recordando que los valores singulares σ_k de $\mathbf{A}$ están relacionados con los valores propios λ_k de $\mathbf{M} = \mathbf{A}^T\mathbf{A}$ mediante $\lambda_k = \sigma_k^2$. **Minimizar** $\mathcal{E}(\mathbf{Y})$ (minimizar λ_k) es equivalente a seleccionar los vectores asociados a los **valores singulares más pequeños** (σ_k).

Los **vectores de embedding** óptimos son los d **vectores columna derechos** de $\mathbf{V}$ que corresponden a los d valores singulares más pequeños, **excluyendo** el valor singular mínimo que es cero ($\sigma_N = 0$).

- **El Vector Cero:** El valor singular σ_N es cero, y su vector derecho asociado $\mathbf{v}_N$ corresponde al vector constante, que es descartado.
- **La Solución:** La matriz de embedding $\mathbf{Y}^*$ se forma tomando los d vectores columna de $\mathbf{V}$ correspondientes a los valores singulares: $\sigma_{N-1}, \sigma_{N-2}, ..., \sigma_{N-d}$.

5.5 Complejidad Computacional y Escalabilidad

El costo dominante de la etapa de embedding reside en la descomposición espectral o SVD.

Operación	Complejidad	Comentarios
Construcción de $\mathbf{A}$	$\mathcal{O}(NK)$	Ensamblaje de la matriz esparsa $\mathbf{A} = \mathbf{I} - \mathbf{W}$.
SVD/Eigen-Decomposition	$\mathcal{O}(N^3)$	Costo de calcular la SVD completa o Eigen-Decomposition para una matriz densa $N \times N$.
Total (Dominante)	$\mathcal{O}(N^3)$	Este costo cúbico limita la aplicación de LLE clásico a $N \lesssim 10^4$.

Referencias

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. A standard comprehensive textbook covering the Manifold Hypothesis and methods like PCA, which set the context for LLE. Springer, 2006.

[2] Ian T. Jolliffe. *Principal Component Analysis*. 2nd. Fundamental text on linear dimensionality reduction, often a comparison point for LLE. Springer, 2002.

[3] Sam T. Roweis and Lawrence K. Saul. “Nonlinear dimensionality reduction by locally linear embedding”. In: *Science* 290.5500 (2000), pp. 2323–2326. DOI: 10.1126/science.290.5500.2323.

[4] Lawrence K. Saul and Sam T. Roweis. “An Introduction to Locally Linear Embedding”. In: *Technical Report*. 2001.

[5] Joshua B. Tenenbaum, Vin De Silva, and John C. Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *Science* 290.5500 (2000), pp. 2319–2323. DOI: 10.1126/science.290.5500.2319.

[6] Zhenyue Zhang and Hongyuan Zha. “Principal manifolds and nonlinear dimensionality reduction via tangent space alignment”. In: *SIAM Journal on Scientific Computing* 26.1 (2004). A related Manifold Learning method (LTSA) often compared to LLE., pp. 313–338. DOI: 10.1137/S106482750343360X.