



UNIVERSIDAD DE CHILE

Minería de Datos

Welcome to the Machine Learning class

Valentin Barriere

Universidad de Chile – DCC

CC5205, Otoño 2024

Disminución de dimensiones

Motivaciones por una dimensión mas baja

Outline : Motivaciones por una dimensión mas baja

Motivaciones por una dimensión mas baja

Selección de atributos

Feature- or Model-based

Feature-based (univariate)

Model-based (scheme-specific)

Reducción de dimensión

Análisis de Componentes

Principales

Otros algoritmos

Selección y Reducción de Atributos

Motivación General

- En muchos problemas de aprendizaje, podemos tener atributos irrelevantes o redundantes.
- Estos atributos pueden afectar negativamente el desempeño de nuestros clasificadores o hacer más costoso el entrenamiento.
- Para abordar este problema, existen dos enfoques principales:
 - **Selección de atributos (supervisado)**: escoger un subconjunto de atributos que sea más útil para la tarea (clasificación, regresión, etc.).
 - **Reducción de dimensionalidad (no supervisado)**: encontrar una proyección de menor dimensión que concentre la información de los datos.

Razones para seleccionar atributos

- Árboles de decisión pueden verse afectados por atributos irrelevantes, pues aunque el árbol intenta escoger atributos relevantes, en la práctica puede “aprender” ruido cuando la profundidad es alta o los datos se fragmentan.
- KNN es muy sensible a atributos irrelevantes, ya que todas las dimensiones participan igual en el cálculo de distancias. Un atributo basura puede distorsionar la cercanía de los vecinos.
- Naïve Bayes tiende a ser robusto a atributos irrelevantes (los ignora porque no modifican significativamente la probabilidad a posteriori), pero sufre con atributos redundantes fuertemente correlacionados.

Razones para seleccionar atributos

- Árboles de decisión pueden verse afectados por atributos irrelevantes, pues aunque el árbol intenta escoger atributos relevantes, en la práctica puede “aprender” ruido cuando la profundidad es alta o los datos se fragmentan.
- KNN es muy sensible a atributos irrelevantes, ya que todas las dimensiones participan igual en el cálculo de distancias. Un atributo basura puede distorsionar la cercanía de los vecinos.
- Naïve Bayes tiende a ser robusto a atributos irrelevantes (los ignora porque no modifican significativamente la probabilidad a posteriori), pero sufre con atributos **redundantes** fuertemente correlacionados.

De una manera general, trata reducir el efecto dañoso de la
curse of dimensionality

Selección de atributos

Outline : Selección de atributos

Motivaciones por una dimensión
mas baja

Selección de atributos

Feature- or Model-based

Feature-based (univariate)

Model-based (scheme-specific)

Reducción de dimensión

Análisis de Componentes
Principales

Otros algoritmos

Outline : Feature- or Model-based

Motivaciones por una dimensión
mas baja

Selección de atributos

Feature- or Model-based

Feature-based (univariate)

Model-based (scheme-specific)

Reducción de dimensión

Análisis de Componentes
Principales

Otros algoritmos

Diferencia entre filtros (feature-based) y envoltura (model-based)

Feature-based o scheme-independent (filtros)

- Evalúan los atributos a partir de **propiedades generales** de los datos (por ejemplo, varianza, correlación, información mutua).
- No utilizan un modelo de clasificación específico para medir la relevancia de cada atributo.
- Son **rápidos** y menos propensos a overfitting, pero **no tienen en cuenta** interacciones complejas entre atributos y la tarea predictiva.

Model-based o scheme-dependent (wrappers)

- Seleccionan atributos **entrenando un clasificador** (o regresor) y evaluando su desempeño sobre cada subconjunto.
- Su objetivo es maximizar la capacidad predictiva de ese modelo.
- Son **computacionalmente más costosos** y pueden sobreajustarse
- Encontran subconjuntos de atributos **más específicos a la tarea**.

Outline : Feature-based (univariate)

Motivaciones por una dimensión
mas baja

Selección de atributos

Feature- or Model-based

Feature-based (univariate)

Correlation Feature Selection

Model-based (scheme-specific)

Reducción de dimensión

Análisis de Componentes
Principales

Otros algoritmos

Métodos Feature-based (Scheme-independent)

Características principales

- No dependen de un modelo específico para medir la relevancia de los atributos ⇒ **Univariate Feature Selection**
- Utilizan **métricas** como entropía, ganancia de información (Information Gain), correlación, chi-cuadrado (χ^2), entre otras.
- Algunos ejemplos populares incluyen:
 - *Information Gain* (basado en entropía)
 - *Mutual Information*
 - *Correlation-based Feature Selection (CFS)*
 - *Low variance*
- Son métodos **rápidos** y sencillos de implementar, aunque **no consideran** la interacción con el modelo de clasificación final.

Mutual Information and χ^2 are useful for sparse data.

Correlation-based Feature Selection (CFS)

- En el esquema **scheme-independent** o de **filtro**, se evalúa el subconjunto de atributos usando una métrica general basada en los datos.
- **CFS** (Correlation-based Feature Selection) busca atributos que:
 - Tengan **alta correlación** con la clase.
 - Tengan **baja correlación** entre ellos (evitar redundancia).
- Se mide la correlación entre atributos categóricos usando *symmetric uncertainty*:

$$\text{SymmUnc}(A, B) = \frac{2 \times (H(A) - H(A|B))}{H(A) + H(B)} = \frac{2 \times IG(A, B)}{H(A) + H(B)},$$

donde $H(\cdot)$ es la entropía e $IG(A, B)$ es la ganancia de información.

$$H(A|B) = H(A, B) - H(B); H(x, y) = -\sum_{x,y} p(x, y) \log p(x, y)$$

Estrategias de búsqueda de subconjuntos

- El número total de subconjuntos de atributos es **exponencial** en el número de atributos (2^n).
- Se utilizan **heurísticas greedy** para no recorrer todo el espacio:
 - **Forward selection:** partir sin atributos y agregarlos uno a uno si mejoran un criterio.
 - **Backward elimination:** partir con todos los atributos y eliminar los que no aporten.
- Métodos más sofisticados:
 - **Best-first search, Beam search, Genetic algorithms**, etc.

Outline : Model-based (scheme-specific)

Motivaciones por una dimensión
mas baja

From Model Training

Performance-based

Selección de atributos

Feature- or Model-based

Reducción de dimensión

Feature-based (univariate)

Análisis de Componentes

Principales

Model-based (scheme-specific)

Otros algoritmos

Selección de atributos basada en importancia

Se puede seleccionar *post-training* los n atributos más importantes de un modelo entrenado mediante la importancia de cada atributo.

- Asigna una **importancia** a cada atributo.
- Se basa en algún atributo específico (por ejemplo, `coef_` o `feature_importances_`) o en un *callable* que devuelva la importancia de cada atributo después de ajustar el modelo.
- Los atributos se consideran **poco importantes** y se eliminan si su valor de importancia es **menor** que un cierto **umbral** (`threshold`).
- Además, el parámetro `max_features` permite fijar un **límite** máximo al número de atributos seleccionados.

Más información en [scikit-learn](#)

Regularización ℓ_1 en la selección de atributos

- ℓ_1 (**Lasso**) induce *parcimonia*: algunos coeficientes se vuelven cero.
- Útil para **selección de atributos** en modelos lineales (regresión o clasificación logística).
- El término de penalización ℓ_1 es:

$$\lambda \sum_j |w_j|$$

- Atributos con coeficientes que se anulan pueden ser descartados sin afectar el modelo.
- Se puede usar con `SelectFromModel` en scikit-learn, usando por ejemplo `Lasso` o `LogisticRegression` con `penalty ℓ_1` .

Selección de atributos por “Wrapper”

- El enfoque **wrapper** evalúa cada subconjunto de atributos entrenando un modelo de aprendizaje y midiendo su **performance** (p.ej. accuracy, F1, AUC).
- Es **computacionalmente más costoso** porque implica entrenar el modelo repetidamente para cada subconjunto evaluado (p.ej., en un *10-fold cross-validation*).
- En una estrategia “greedy”, cada nuevo atributo a probar se evalúa múltiples veces, lo que puede llegar a ser $O(m^2)$ o incluso $O(2^m)$ en el peor de los casos (búsqueda exhaustiva).
- Se comporta muy bien con modelos como Naive Bayes, pero el costo puede ser prohibitivo con un número grande de atributos.

Recursive Feature Elimination (RFE)

Concepto general

- RFE parte de un estimador externo que asigna **pesos** a los atributos (por ejemplo, los coeficientes de un modelo lineal o los *feature importances* de un árbol/random forest).
- El objetivo es **seleccionar atributos** considerando conjuntos cada vez más pequeños.
- Procedimiento:
 1. Entrenamos el estimador con el *conjunto inicial* de atributos.
 2. Obtenemos la **importancia** de cada atributo (por ejemplo, leyendo `coef_` o `feature_importances_`).
 3. Se **descartan** los atributos menos importantes (poda).
 4. Se repite el proceso de forma **recursiva** sobre el nuevo subconjunto reducido hasta llegar al número deseado de atributos.

Reducción de dimensión

Outline : Reducción de dimensión

Motivaciones por una dimensión
mas baja

Selección de atributos

Feature- or Model-based

Feature-based (univariate)

Model-based (scheme-specific)

Reducción de dimensión

Análisis de Componentes
Principales

Otros algoritmos

Intereses de la Reducción de Dimensión

- **Simplificar los datos:** Menos atributos implican menos ruido y mayor capacidad de generalización.
- **Reducir el costo computacional:** Entrenar y usar modelos con muchos atributos puede ser muy costoso.
- **Visualización:** Proyectar datos de alta dimensión a 2D o 3D permite interpretaciones gráficas y facilita la comprensión.
- **Evitar la maldición de la dimensionalidad:** Métodos basados en distancias sufren mucho con dimensiones muy altas.
- **Comprimir los datos:** Permite de reducir el espacio en memoria de una base de datos

Outline : Análisis de Componentes Principales

Motivaciones por una dimensión mas baja

Selección de atributos

Feature- or Model-based

Feature-based (univariate)

Model-based (scheme-specific)

Reducción de dimensión

Análisis de Componentes Principales

Introducción

Motivación y Ejemplo

Vectores Propios y Covarianza

Principio

Ejemplos visuales

Otros algoritmos

PCA: Análisis de Componentes Principales

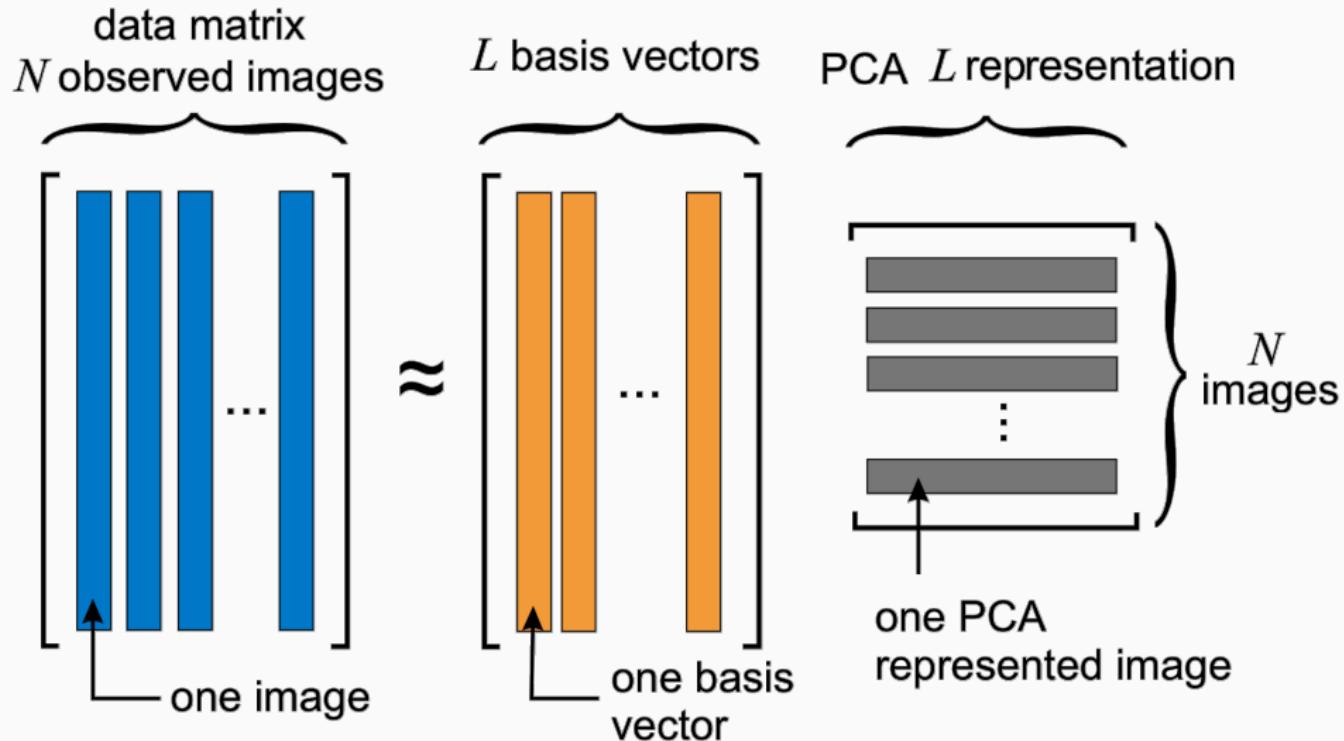
Utiliza los vectores propios de la matriz de covarianza de los datos para encontrar las representaciones que son más generales



Motivación

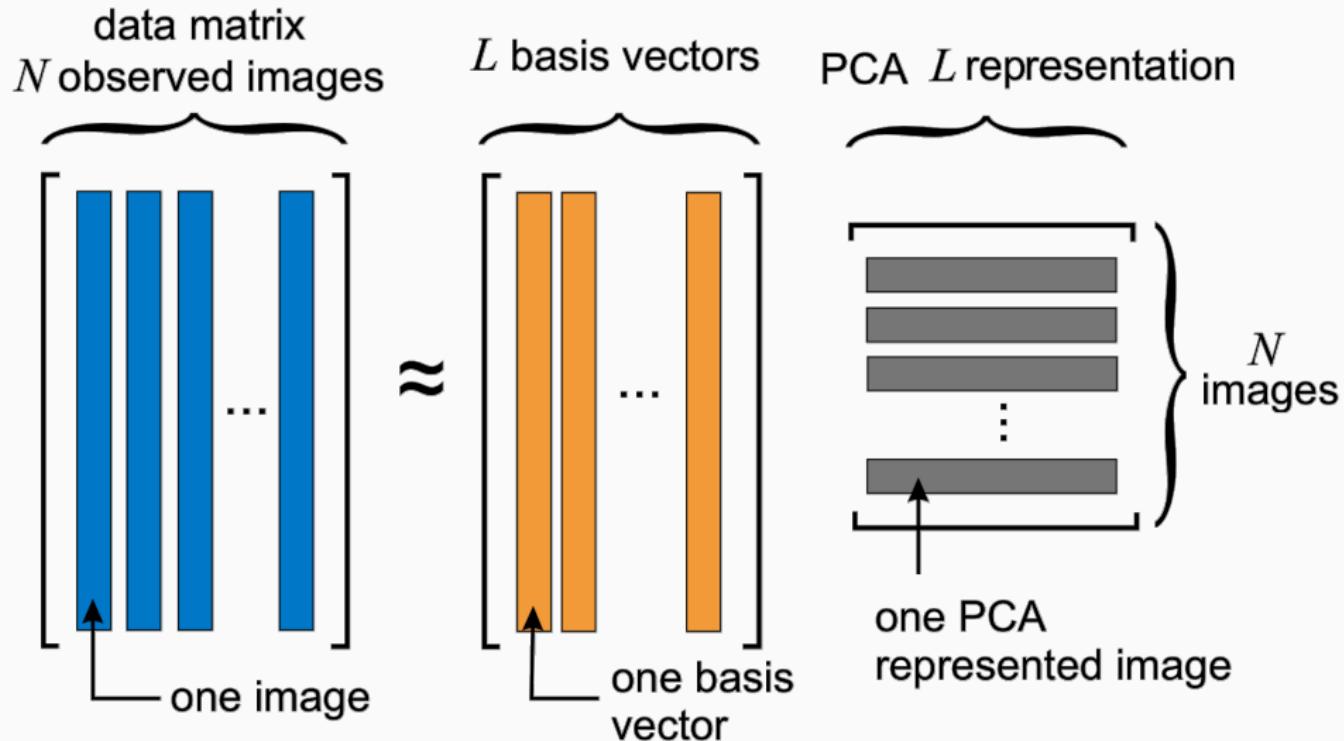
- En estadística, el PCA es un método para simplificar una base de datos multidimensional en una dimensión más baja. En otras palabras, **se reduce la dimensión del espacio de los descriptores**.
- Esta reducción de dimensión se realiza con fines de análisis, visualización y compresión de datos.
- El PCA representa los datos utilizando un **nuevo sistema de coordenadas en el que los vectores base siguen los modos de mayor varianza en los datos**. En otras palabras, los vectores que representan la matriz de covarianza.
- Por lo tanto, se calcula una nueva base de vectores para un conjunto de datos específico.

PCA: Ilustración



Cada dato (aca imagen) se representa en esta nueva base mas chica

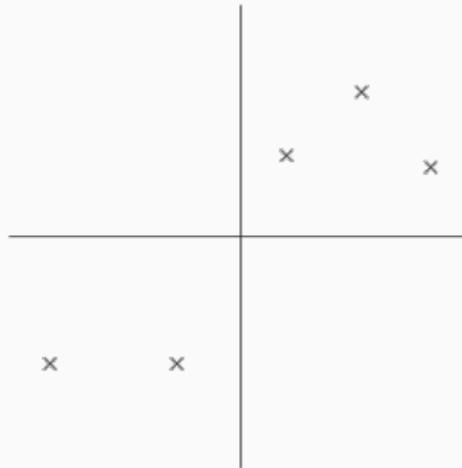
PCA: Ilustración



Cada dato (aca imagen) se representa en esta nueva base mas chica
⇒ se representa con menos coordinados.

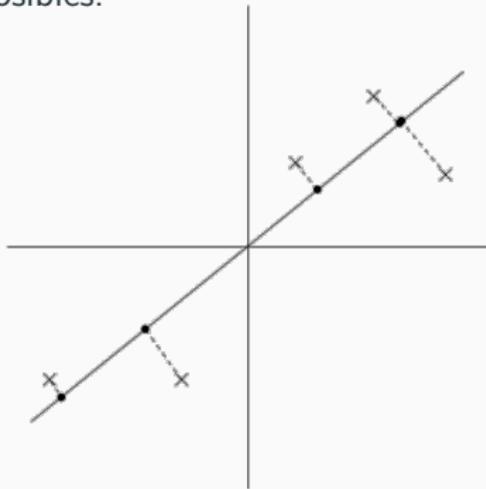
PCA: Motivación

- **Redundancia de atributos:** Supongamos que tenemos atributos casi linealmente dependientes, como la velocidad en km/h y en mph.
- Esto puede considerarse como “información repetida”: a menudo queremos eliminar esa redundancia y quedarnos con un único atributo más representativo.
- **Objetivo de PCA:** Encontrar una proyección de los datos donde la mayoría de la varianza sea capturada con la menor cantidad de componentes posibles.



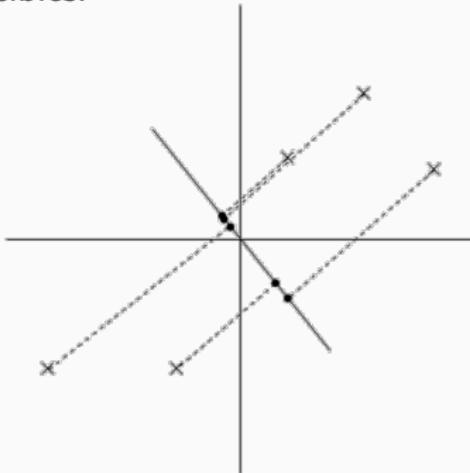
PCA: Motivación

- **Redundancia de atributos:** Supongamos que tenemos atributos casi linealmente dependientes, como la velocidad en km/h y en mph.
- Esto puede considerarse como “información repetida”: a menudo queremos eliminar esa redundancia y quedarnos con un único atributo más representativo.
- **Objetivo de PCA:** Encontrar una proyección de los datos donde la mayoría de la varianza sea capturada con la menor cantidad de componentes posibles.



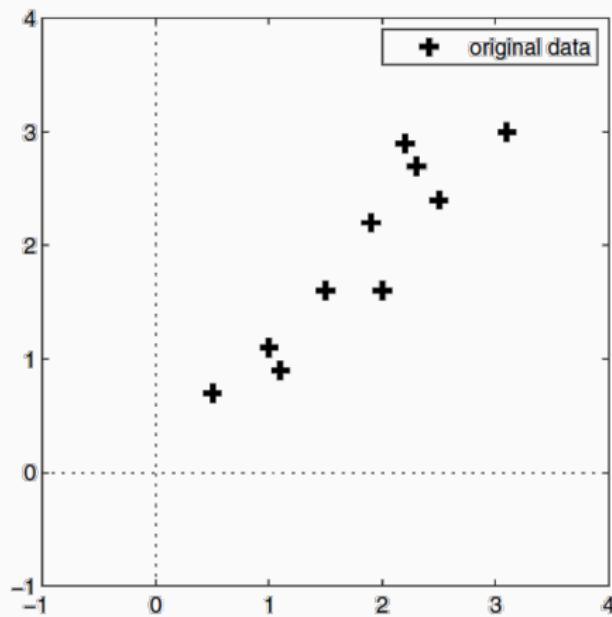
PCA: Motivación

- **Redundancia de atributos:** Supongamos que tenemos atributos casi linealmente dependientes, como la velocidad en km/h y en mph.
- Esto puede considerarse como “información repetida”: a menudo queremos eliminar esa redundancia y quedarnos con un único atributo más representativo.
- **Objetivo de PCA:** Encontrar una proyección de los datos donde la mayoría de la varianza sea capturada con la menor cantidad de componentes posibles.



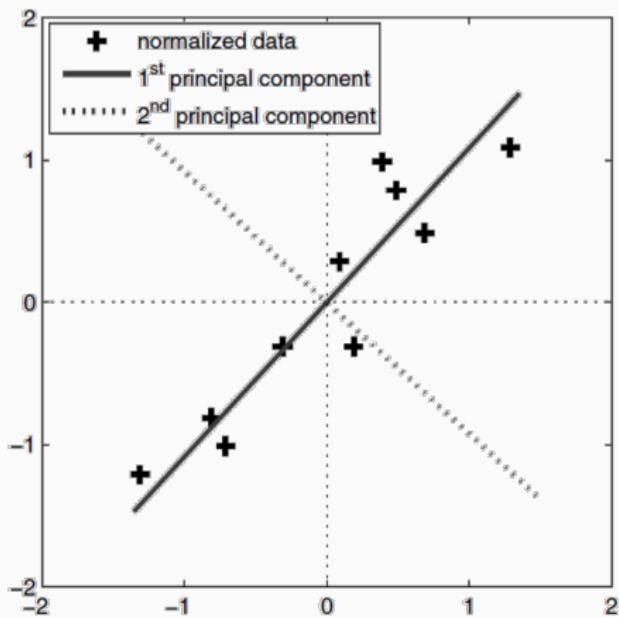
PCA: Visualización de las Transformaciones

- El dataset inicial



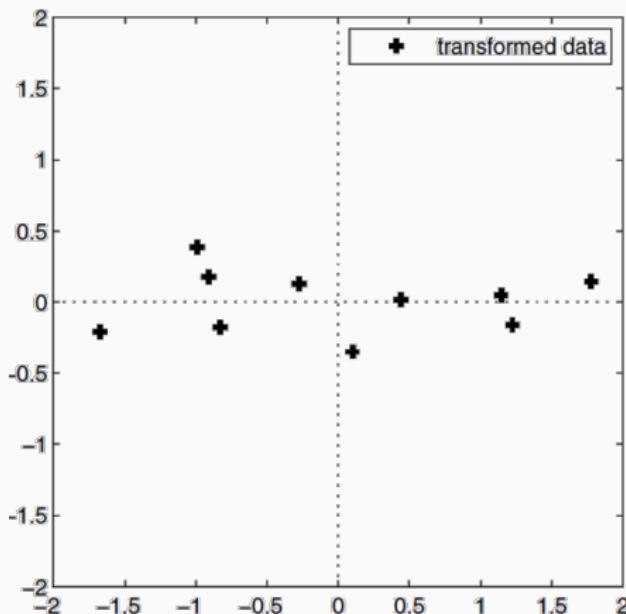
PCA: Visualización de las Transformaciones

- El dataset inicial
- Los datos normalizados con los ejes de la PCA



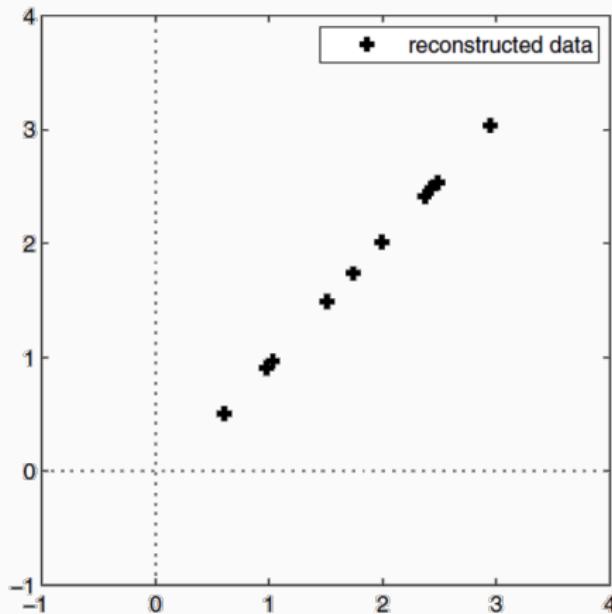
PCA: Visualización de las Transformaciones

- El dataset inicial
- Los datos normalizados con los ejes de la PCA
- Los datos transformados y normalizados (con los nuevos ejes)



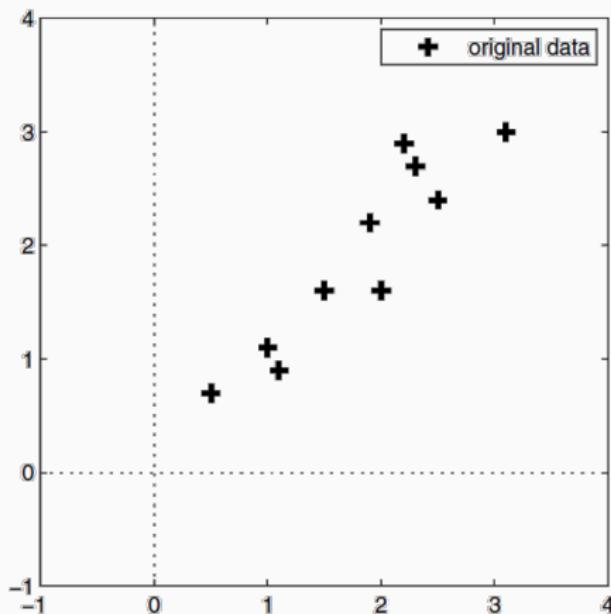
PCA: Visualización de las Transformaciones

- El dataset inicial
- Los datos normalizados con los ejes de la PCA
- Los datos transformados y normalizados (con los nuevos ejes)
- Los datos reconstruidos por la PCA (con perdida de información)



PCA: Visualización de las Transformaciones

- El dataset inicial
- Los datos normalizados con los ejes de la PCA
- Los datos transformados y normalizados (con los nuevos ejes)
- Los datos reconstruidos por la PCA (con perdida de información)



Recordatorio sobre vectores y valores propios

- Sea A una matriz cuadrada de tamaño $n \times n$
- Los **vectores propios** son las soluciones de la ecuación:

$$A\mathbf{u} = \lambda\mathbf{u}$$

donde λ se llama el **valor propio**

- El valor de λ es significativo en términos de su importancia para representar el conjunto de vectores **imagen** obtenidos a través de la matriz A

Si $\det(A) \neq 0$ entonces $\exists U = \begin{pmatrix} | & | & | \\ \mathbf{u}_1 & \dots & \mathbf{u}_n \\ | & | & | \end{pmatrix}$ y $D = \begin{pmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_d \end{pmatrix}$

tal que $A = UDU^{-1}$ tiene como vectores y valores propios \mathbf{u}_i y λ_i

Recordatorio sobre la Covarianza

- **Definición:** La covarianza entre dos variables aleatorias X e Y se define como:

$$\text{cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])].$$

- **Interpretación:**

- $\text{cov}(X, Y) > 0$ indica que cuando X aumenta, Y tiende a aumentar también (relación directa).
- $\text{cov}(X, Y) < 0$ indica que cuando X aumenta, Y tiende a disminuir (relación inversa).
- $\text{cov}(X, Y) = 0$ sugiere que no hay relación lineal aparente entre X y Y .

- **Matriz de Covarianza:** Para un conjunto de variables, se construye una matriz cuadrada donde cada entrada es la covarianza entre un par de variables.

PCA: Aproximación de los Datos

Si aproximamos los datos $\mathbf{X} = (\mathbf{X}_i)_i$ mediante un espacio de menor dimensión L , entonces el error de aproximación sería mínimo para un espacio L con base

$$(\mathbf{b}_i)_{i=1..L} = \arg \max_{(\mathbf{b}_i)_{i=1..L}} \sum_{i=1}^L \mathbf{b}_i^T \text{cov}(\mathbf{X}) \mathbf{b}_i$$

Solución

Eso corresponde a los **vectores propios de $\text{cov}(\mathbf{X})$** ¹ con los valores propios más grandes.

Intuición: Maximizar la cantidad proyectada sobre u

$$\max_{\|u\|=1} \sum_{i=1}^d (\mathbf{X}_{0,i}^T \cdot u) = (\mathbf{X}_0 u)^T (\mathbf{X}_0 u) = u^T (\mathbf{X}_0^T \mathbf{X}_0) u = \max_{\|u\|=1} u^T \Sigma u$$

Este se resuelve simplemente con un Lagrangiano!

¹ $\text{cov}(\mathbf{X}) = (\mathbf{X} - \mu_{\mathbf{X}})^T (\mathbf{X} - \mu_{\mathbf{X}})$

PCA: Normalización por centrado de las muestras

Tomemos nuestra base de datos $\mathbf{X} = \begin{pmatrix} | & | & | \\ \mathbf{x}_1 & \dots & \mathbf{x}_n \\ | & | & | \end{pmatrix}$

- Para no tener en cuenta los sesgos en el cálculo, **el procedimiento debe aplicarse a la base de datos normalizada \mathbf{X}_0**
- Se resta la media empírica a cada columna de \mathbf{X} :

$$\mathbf{X}_0 = \begin{pmatrix} | & | & | \\ \mathbf{x}_1 - \mu_1 & \dots & \mathbf{x}_n - \mu_n \\ | & | & | \end{pmatrix}, \text{ donde } \mu_i = \frac{1}{d} \sum_{k=1}^d \mathbf{x}_i^{(k)}$$

También podemos elegir normalizar con la varianza:

$$\mathbf{X} = \begin{pmatrix} | & | & | \\ \frac{\mathbf{x}_1 - \mu_1}{\sigma(\mathbf{x}_1)} & \dots & \frac{\mathbf{x}_n - \mu_n}{\sigma(\mathbf{x}_n)} \\ | & | & | \end{pmatrix}, \text{ donde } \sigma_i = \sqrt{\frac{1}{d} \sum_{k=1}^d (\mathbf{x}_i^{(k)} - \mu_i)^2}$$

PCA: Principio

- Cada uno de los datos \mathbf{X}_i del conjunto se considera como un vector en el espacio de los descriptores (puede ser un sonido, una imagen)
- Buscamos encontrar vectores que sean más representativos de estos datos:
 - El espacio de datos tiene una base ortogonal natural que se puede utilizar para esto.
 - Cada uno de los datos se puede expresar en función de los vectores de esta base ortogonal.
- Esta base es la base de los vectores propios de la matriz de covarianza de los datos centrados $\Sigma = \mathbf{X}_0^T \mathbf{X}_0$

Valor propio

Cada uno de los vectores de esta nueva base puede clasificarse según su propensión a representar la mayoría de los $\mathbf{X}_{0,i}$. Esta propensión corresponde al valor propio asociado al vector de la nueva base.

1. Crear una matriz de datos $\mathbf{X} = \begin{pmatrix} | & | & | \\ \mathbf{x}_1 & \dots & \mathbf{x}_n \\ | & | & | \end{pmatrix}$
2. Normalizar los datos:
$$\mathbf{x}_0 = \mathbf{X} - \mu, \text{ donde } \mu_i = \frac{1}{d} \sum_{k=1}^d \mathbf{x}_i^{(k)}$$
3. Calcular la matriz de covarianza $\Sigma = \mathbf{x}_0^T \mathbf{x}_0$
4. Encontrar los vectores propios u_k y valores propios λ_k de Σ
5. Tomar los L vectores propios tales que $\frac{\sum_k^L \lambda_k}{\sum_k^n \lambda_k} > 0.99$ para reconstruir el 99% de los datos
6. Recuperar las representaciones de las imágenes en \mathbb{R}^L (si se reduce la dimensión)

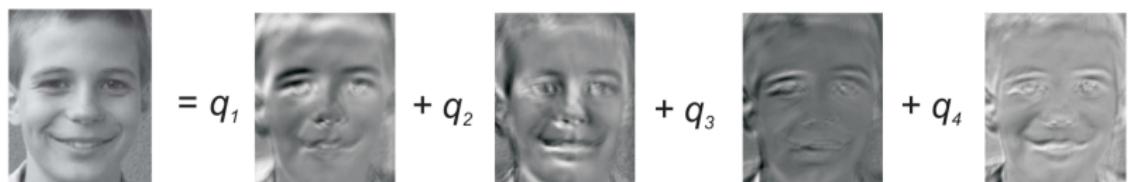
PCA: Ejemplo con 32 imágenes



Visualización y reconstrucción

- Reconstrucción de la imagen desde nuestros 4 vectores base $\mathbf{u}_i, i = 1 \dots 4$.
- La combinación lineal se calculó de la siguiente manera:

$$0,078\mathbf{u}_1 + 0,062\mathbf{u}_2 - 0,182\mathbf{u}_3 + 0,179\mathbf{u}_4$$



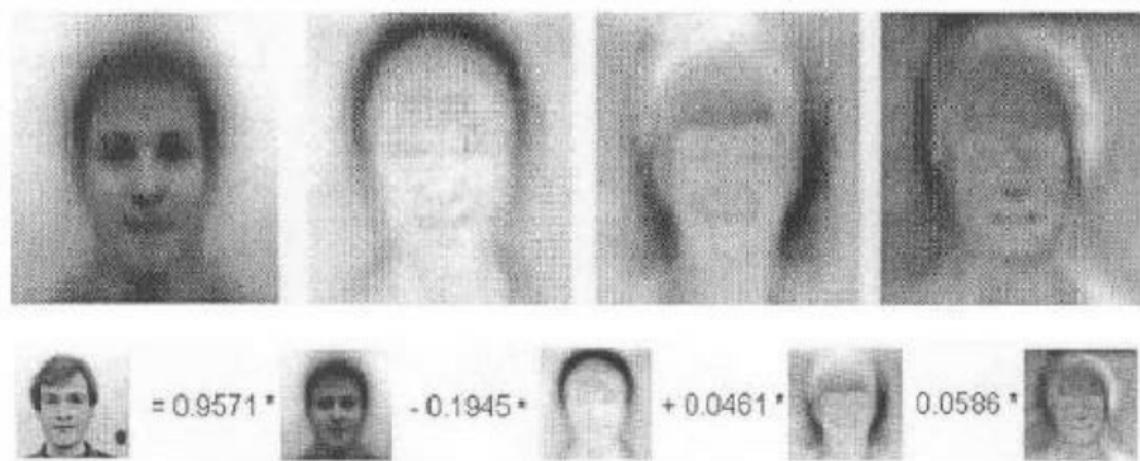
Reconstrucción: 4 componentes



Reconstrucción: Datos brutos



PCA: Otro ejemplo



Outline : Otros algoritmos

Motivaciones por una dimensión
mas baja

Selección de atributos

Feature- or Model-based

Feature-based (univariate)

Model-based (scheme-specific)

Reducción de dimensión

Análisis de Componentes
Principales

Otros algoritmos

Otros algoritmos de reducción de dimensión

- **Multidimensional Scaling (MDS)**: Mapea los objetos a una dimensionalidad menor, buscando mantener las distancias relativas entre objetos
- **t-Distributed Stochastic Neighbor Embedding (TSNE)**: Modela cada objeto de alta dimensión por un punto de dos o tres dimensiones de tal manera que los objetos similares son modelados por puntos cercanos y los objetos diferentes son modelados por puntos distantes con alta probabilidad. Mas explicaciones [aca](#)
- **ICA (Independent Component Analysis)**:
- **Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP)**: Basado Geometría Riemanniana y topología algebraica. Mas explicaciones [aca](#)
- **Auto-encoder**: red neuronal no supervisada que comprime los datos (encoder) y la expande nuevamente (decoder) para reconstruir la entrada minimizando el error, aprovechando así su capa latente como reducción de dimensionalidad.

Conclusiones

- **Selección de atributos** (filter o wrapper) reduce la complejidad del modelo y puede mejorar rendimiento, sobre todo para clasificadores sensibles a atributos irrelevantes.
- **Reducción de dimensionalidad** (PCA, Autoencoders, MDS, TSNE, etc.) agrupa la información en menos dimensiones, pudiendo mejorar visualización y la robustez de métodos basados en distancias.
- Siempre debemos balancear la **ganancia en simplicidad** con la **pérdida de información** al eliminar/proyectar atributos.

Questions?

References i