Ricardo Aler Mur

SELECCIÓN Y GENERACIÓN DE ATRIBUTOS-I

En esta clase se habla de una parte importante del preprocesado de datos: la selección y generación de atributos. La selección de atributos consiste en elegir un subconjunto de los atributos más apropiados, mientras que la generación consiste en generar (y seleccionar entre ellos) nuevos atributos a partir de los ya existentes.

SELECCIÓN DE ATRIBUTOS

Se tratan los siguientes temas:

- ¿Por qué hacer selección de atributos? Principalmente por tres razones: la existencia de atributos irrelevantes, la existencia de atributos redundantes y la maldición de la dimensionalidad. La primera puede producir problemas de sobreaprendizaje además de hacer más confusos los modelos resultantes. La segunda es nociva para ciertos algoritmos de aprendizaje. La tercera es una cuestión a tener en cuenta cuando hay pocos datos en relación a la presencia de muchos atributos.
- Una cuestión importante en selección de atributos es que en algunos problemas puede ocurrir que algunos atributos no estén correlacionados con la clase por separado pero si cuando actúan juntos, por lo que el objetivo último de la

selección es encontrar el subconjunto de atributos más apropiado.

- Para definir un método de selección de atributos es necesario definir un espacio de búsqueda y un método de evaluación de la calidad de los subconjuntos.
- Se clasifican los métodos de selección en dos tipos principales: ranking y selección de subconjuntos (subsets)

GENERACIÓN DE ATRIBUTOS:

Se tratan los siguientes temas:

- Construcción de nuevos atributos mediante Principal Component Analysis o PCA. Se trata de identificar un primer componente que explique la mayor cantidad posible de varianza, un segundo componente que explique la siguiente mayor cantidad de varianza y así sucesivamente. Dado que este método ordena los atributos, también se puede utilizar como método de selección.
- Se intenta transmitir la idea de que PCA consiste en intentar determinar si un conjunto de datos se puede expresar mediante una dimensionalidad inferior al número real de atributos en el problema. Se pone como ejemplo un conjunto de datos en un plano, pero que dicho plano está "embebido" en un espacio de muchas mas dimensiones.
- Se destacan tres aspectos de PCA: que se trata de una transformación lineal de los atributos originales, que es una transformación no supervisada (con lo que hay que tener cuidado para problemas de clasificación) y que cuando el número de atributos es muy grande, el método puede ser lento.

- Para resolver el problema de la lentitud de PCA, se introduce el método de proyecciones aleatorias (Random Projections), que para un número de atributos grande obtiene resultados similares a PCA, con un menor esfuerzo computacional.

Ricardo Aler Mur

SELECCIÓN DE ATRIBUTOS

Fases del análisis de datos

- Recopilación de los datos (tabla datos x atributos)
- Preproceso:
  - De los datos
  - De los atributos
- Generación del modelo (clasificador, …) y de la estimación de la calidad del modelo (estimación del porcentaje de aciertos futuro por validación cruzada)
- Despliegue y uso del modelo

PREPROCESO

- De atributos:
  - Normalización
  - Creación de dummy variables a partir de variables discretas
  - Imputación (qué hacer con los valores faltantes)
  - **Selección de atributos**
  - **Creación de atributos**
- De datos:
  - Eliminación de outliers (datos raros)
  - Muestreo para manejar un conjunto de datos más pequeño pero representativo
  - Muestreo para equilibrar las dos clases en un problema de muestra desbalanceada

Selección de atributos

- Algunos atributos pueden ser **redundantes** (como "salario" y "categoría social")
  - Hacen más lento el proceso de aprendizaje (Ej: C4.5 $O(m*n^2)$ SVM $O(m*n)$)
  - Pueden confundir a algunos clasificadores (como el Naive Bayes)
- Otros son **irrelevantes** (como el DNI para predecir si una persona va a devolver un crédito)
  - En algunos estudios, un solo atributo irrelevante (aleatorio) perjudica un 5% o 10% al clasificador (C4.5 en este caso)
- **Maldición de la dimensionalidad**:
  - El número de datos necesarios puede crecer exponencialmente con el número de dimensiones
  - El exceso de atributos puede llevar a sobreaprendizaje, pues incrementa la complejidad del modelo en relación al número de datos disponibles
- En ocasiones es útil tener el conocimiento de qué atributos son relevantes para una tarea
- Cuantos menos atributos, mas fácil de interpretar es el modelo
- Algunos algoritmos (como C4.5 árboles de decisión) son capaces de descartar atributos. Pero también existen algoritmos de selección de atributos que se pueden usar para preprocesar los datos

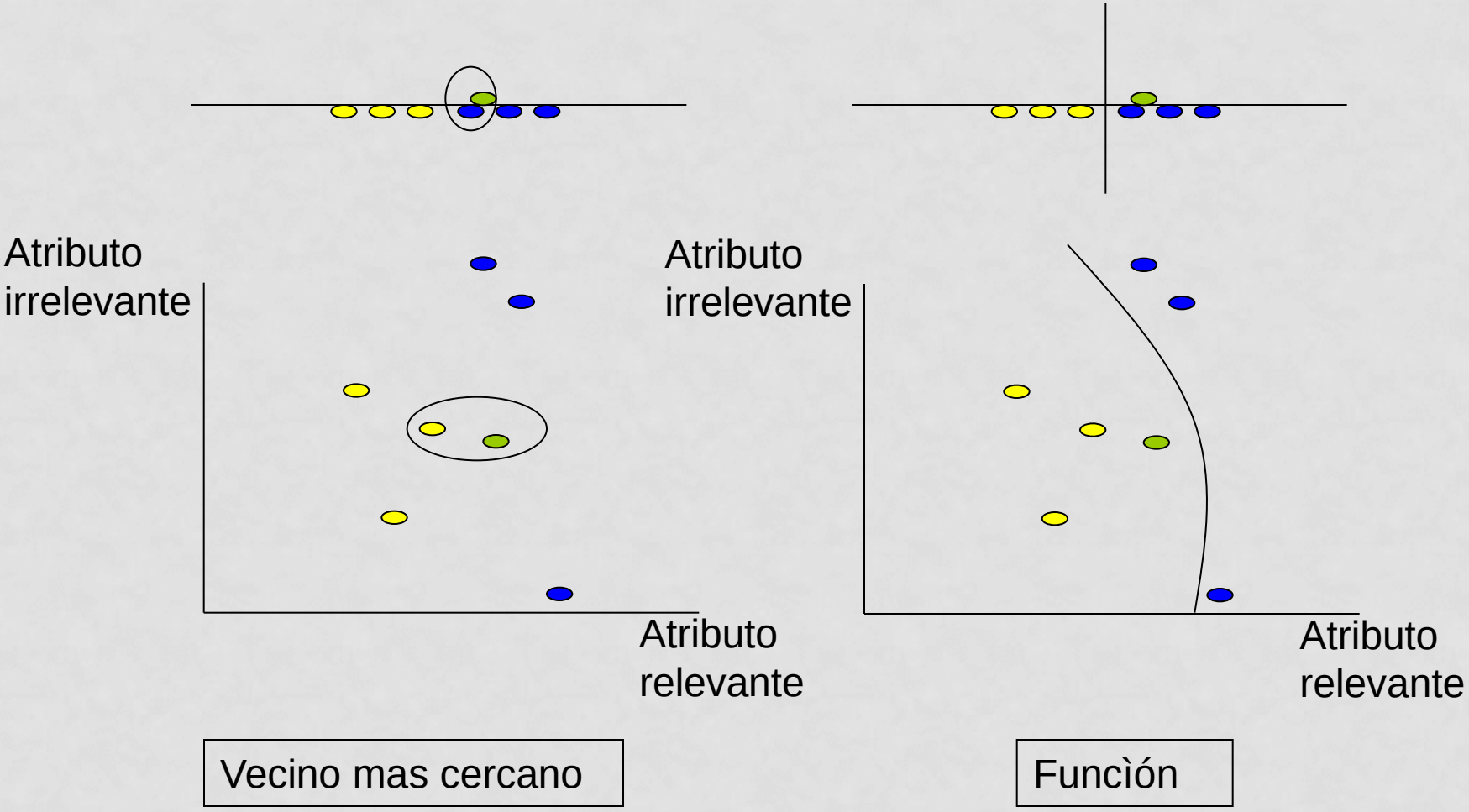
Atributos redundantes

- Ejemplo en Naive Bayes, el cual supone que los atributos son independientes
- Pr(si/cielo = sol, temperatura = frío, humedad = alta, viento = si)

  = k* pr(cielo = sol/si) * pr(humedad = alta /si) * pr(temperatura = alta /si)* pr(viento = si /si) * Pr(si)
- Supongamos que temperatura y humedad son completamente redundantes (o sea, temperatura=humedad)
  - No es cierto que lo sean, sólo lo suponemos
  - Entonces es como si el atributo humedad se le tuviera en cuenta dos veces, frente a los demás que sólo se les cuenta una:

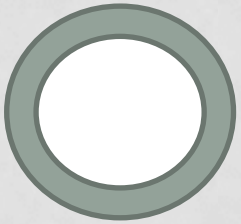
  = k* pr(cielo = sol/si) * pr(humedad = alta /si) * pr(humedad = alta /si)* pr(viento = si /si) * Pr(si)

Atributos irrelevantes



INTUICIÓN DE MALDICIÓN DE DIMENSIONALIDAD

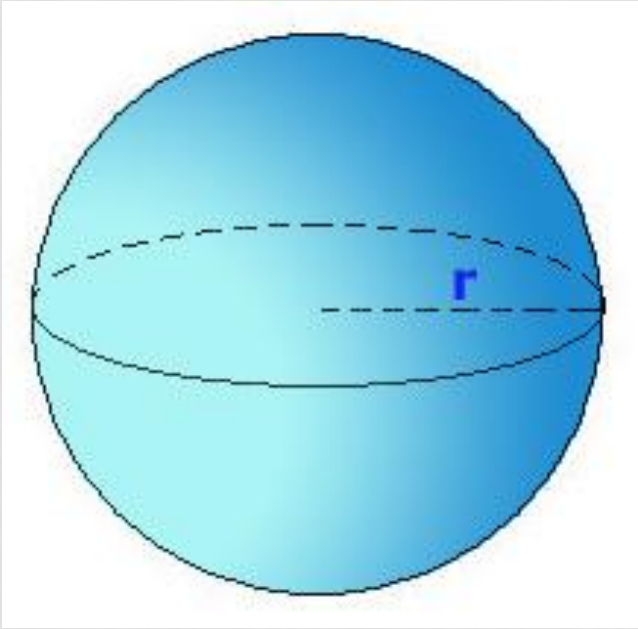
- No tenemos una intución desarrollada para saber lo que pasa en muy altas dimensionalidades
- Supongamos que tenemos dos circunferencias concéntricas, una de radio 1, y otra de radio 0.9
- ¿Cuál es la superficie que queda entre ambas circunferencias?
- Superficie del primer círculo = $pi*1^2$
- Superficie del segundo círculo = $pi*0.9^2$
- Diferencia, en proporción al primer círculo
  = $pi*(1^2-0.9^2) / (pi*1^2) = 1^2-0.9^2 = 0.19 => 20\%$



- Sabiendo que el volumen de una hiperesfera en n-dimensiones es
  = $k(n) * r^n$
- ¿Cuál sería la proporción de espacio que queda entre las dos hiperesferas (radios 1 y 0.9) pero en un espacio de 50 dimensiones?

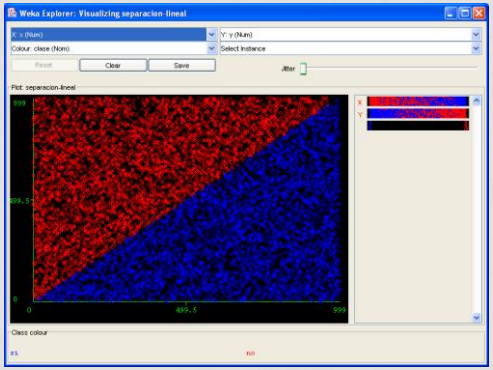
Maldición de la dimensionalidad

- Nótese que las áreas crecen exponencialmente con el número de dimensiones.
- Ejemplo, superficie de una esfera:
  - 2D: $2\pi r$
  - 3D: $4\pi r^2$
  - 4D: $2\pi^2 r^3$
  - dD: $O(r^{d-1})$



Maldición de la dimensionalidad en un clasificador lineal

- Sea un problema de clasificación biclase con **1000 atributos**
- Disponemos de un algoritmo que genera clasificadores lineales
- Supongamos que tenemos **1001 datos de entrenamiento** (y por ejemplo 10000 para test)
- ?Cuál será el porcentaje de aciertos en entrenamiento?
- ?Cuál será el porcentaje de aciertos en test?



$$A_1*X_1 + A_2*X_2 + A_3*X_3 + \ldots + A_{1000}*X_{1000} > A_0$$

Ventajas de la selección de atributos (feature selection)

- Aliviar el efecto de la maldición de la dimensionalidad
- Mejorar la capacidad de generalización (eliminando atributos irrelevantes y redundantes)
- Acelerar el proceso de aprendizaje (aunque se añade el coste de la selección en el preproceso)
- Mejorar la interpretabilidad del modelo (al reducir la complejidad del modelo)

Idea importante

- En ocasiones, dos atributos por separado no dan información, pero juntos sí
- Ejemplo:
  - Sea un problema de clasificación de textos en dos clases "informática" y "filosofía"
  - Sean los atributos booleanos "inteligencia" y "artificial", que son ciertos si esas palabras aparecen en el texto y falsos en caso contrario
  - Por separado no permiten distinguir entre informática y filosofía:

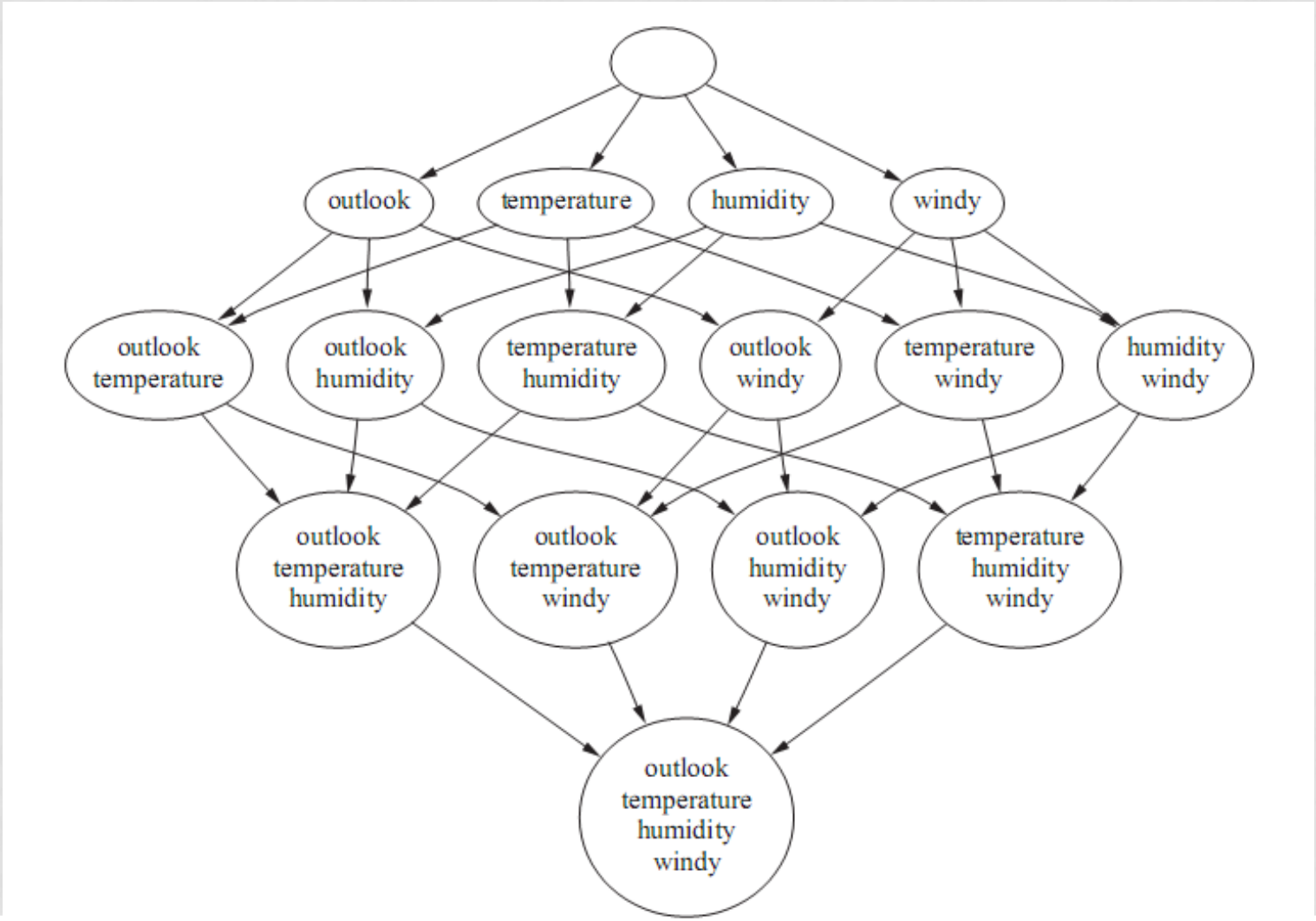
    IF inteligencia=si **THEN** ?; **IF** artificial=si **THEN** ?
  - Pero juntos sí:

    IF inteligencia=si **Y** artificial=si **THEN** "informática"
- Por tanto, el objetivo último de la selección de atributos es encontrar el **subconjunto** mínimo de atributos que hace óptima la predicción
- Ejemplo: datos xor

Búsqueda exhaustiva

- El método mas preciso sería la búsqueda exhaustiva
- Supongamos que tenemos 4 atributos A, B, C, D
- Sería necesario comprobar la validez de todos los posibles subconjuntos (2^4=16): {A, B, C, D}, {A, B, C}, {A, B, D}, {B, C, D}, {A, C, D}, {A, B}, {A, C}, ..., {A}, {B}, {C}, {D}
- En general, el método es poco práctico: 2^n posibles subconjuntos

Búsqueda en el espacio de subconjuntos de atributos



TAXONOMÍA DE MÉTODOS DE SELECCIÓN DE ATRIBUTOS

	Filter	Wrapper
Ranking (atributos individuales)	Information Gain, Información Mútua, Chi-square	
Subset selection	Correlation Feature Selection (CFS)	

RELIEF

Ranking (evaluación y ordenación de atributos de manera individual y eliminación de los menos valorados)

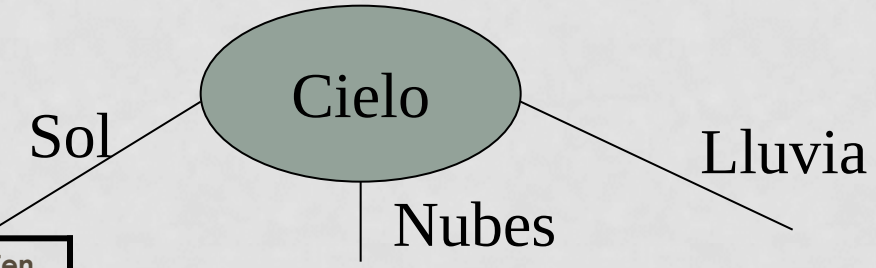
Subset selection (búsqueda del subconjunto de atributos más relevante)

- Métodos de búsqueda:
  - Greedy stepwise:
    - Hacia adelante
    - Hacia atrás
  - Mejor primero (best first)
  - Búsqueda genética

Ranking

- Dado unos atributos A_1, A_2, ..., A_n, se evalúa cada A_i de manera independiente, calculando medidas de correlación del atributo con la clase
- Un atributo A_1 está correlacionado con la clase, si conocer su valor implica que podemos predecir la clase con cierta probabilidad
  - Por ejemplo, el sexo de una persona está correlacionado (de momento) con que le guste el fútbol. Su DNI no lo está
  - Por ejemplo, el salario de una persona está correlacionado con el hecho de que vaya a devolver un crédito
- Criterios para evaluar a los atributos:
  - Entropía (information gain), como en los árboles de decisión
  - Chi-square
  - Mutual information
  - …
- Una vez evaluados y ordenados, se quitan los k peores

Entropía / Information Gain para ordenar atributos



C	T	H	V	Ten
Sol	85	85	No	No
Sol	80	90	Si	No
Sol	72	95	No	No
Sol	69	70	No	Si
Sol	75	70	Si	Si

C	T	H	V	Ten
Nubes	83	86	No	Si
Nubes	64	65	Si	Si
Nubes	72	90	Si	Si
Nubes	81	75	No	Si

C	T	H	V	Ten
Lluvia	70	96	No	No
Lluvia	68	80	No	Si
Lluvia	75	80	No	Si
Lluvia	65	70	Si	No
Lluvia	71	91	Si	No

"3 No, 2 Si"

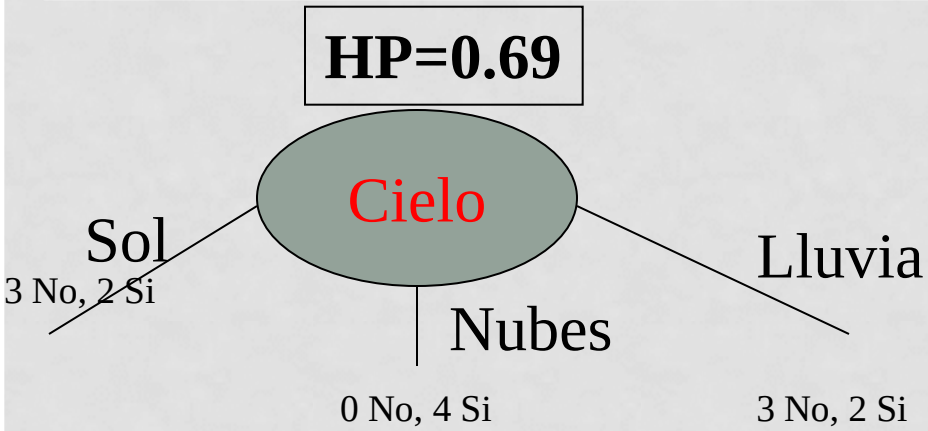
"0 No, 4 Si"

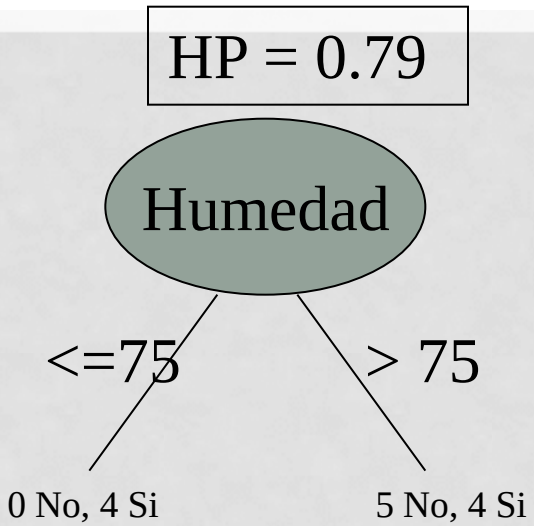
"3 No, 2 Si"

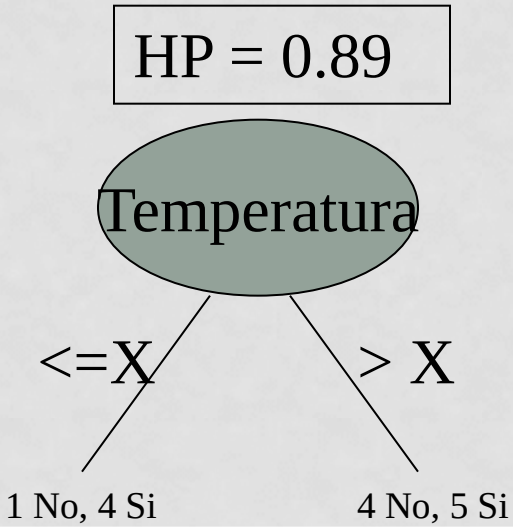
$$H(P) = -(p_{si}\log_2(p_{si}) + p_{no}\log_2(p_{no}))$$

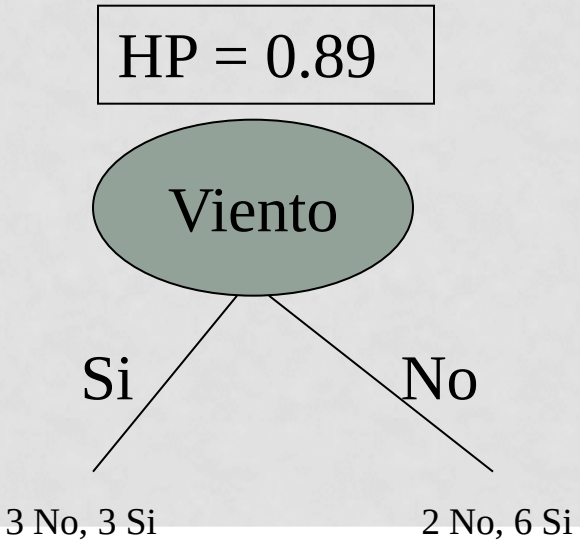
$$p_{no} = (1 - p_{si})$$

Atributos ordenados (ranking) por entropía









Ej: Correlación a través de mutual information

Recordar que si x e y son independientes, p(x,y)=p(x)*p(y)

$$I(x,y) = \sum_i \sum_j p(x=i, y=j) \log \left[\frac{p(x=i, y=j)}{p(x=i)p(y=j)} \right]$$

- i son los valores del atributo x, j son los valores de la clase y
- I(x,y)=0 si x e y son independientes (log(1) = 0)
- I(x,y)>= 0 (cuanto mas correlacionadas están las variables, mas positiva es la información mútua)

Ranking

- Ventajas: es rápido
- Desventajas:
  - No elimina atributos redundantes
  - No detecta atributos que funcionan bien de manera conjunta, pero mal de manera separada. De hecho, descartaría esos atributos.
    - Ej: las aparición de las palabras “inteligencia” y “artificial” no está excesivamente correlacionado por separado con textos de informática, pero juntas su correlación se incrementa notablemente

TAXONOMÍA DE MÉTODOS DE SELECCIÓN DE ATRIBUTOS

	Filter	Wrapper
Ranking (atributos individuales)	Information Gain, Información Mútua, Chi-square	
Subset selection	Correlation Feature Selection (CFS)	

RELIEF

Ranking (evaluación y ordenación de atributos de manera individual y eliminación de los menos valorados)

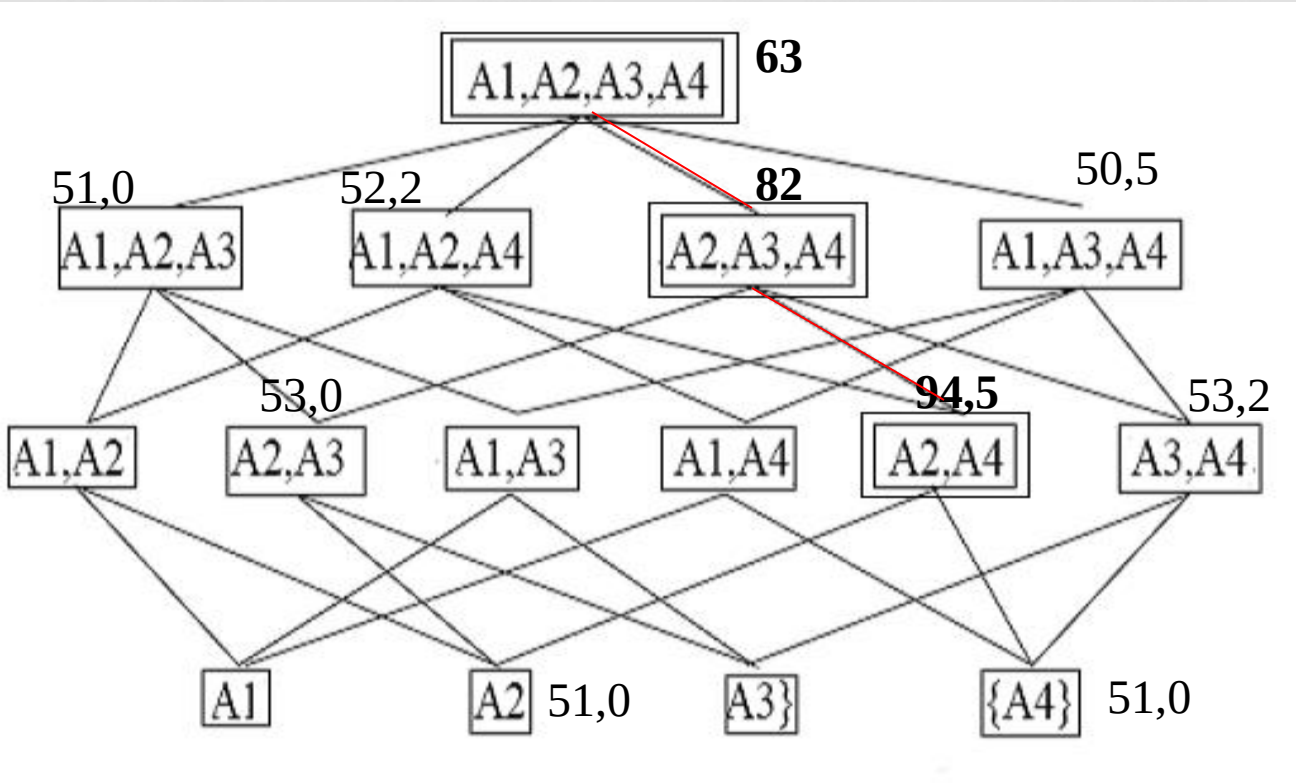
Subset selection (búsqueda del subconjunto de atributos más relevante)

- Métodos de búsqueda:
  - Greedy stepwise:
    - Hacia adelante
    - Hacia atrás
  - Mejor primero (best first)
  - Búsqueda genética

Selección de subconjuntos (subset selection)

- Estos métodos recorren un espacio de búsqueda de subconjuntos de atributos, evaluando **subconjuntos** completos de atributos
- No se recorre el espacio entero (eso sería búsqueda exhaustiva), sino sólo aquellos subconjuntos más prometedores
- Se evalúa el subconjunto de manera conjunta

- Tipos:
  - CFS
  - Wrapper



Selección de subconjuntos

- Hay que definir:
  - **Una manera de moverse por el espacio de búsqueda** de subconjuntos de atributos:
    - Hacia adelante
    - Hacia detrás
    - …
  - Una manera (medida) de evaluar subconjuntos de atributos

Selección de subconjuntos

- Hay que definir:
  - Una manera de moverse por el espacio de búsqueda de subconjuntos de atributos:
    - Hacia delante
    - Hacia detrás
    - …
  - **Una manera (medida) de evaluar subconjuntos de atributos**

Evaluación de subconjuntos: Correlation Feature Selection (CFS)

- El método *CFS* evalúa un subconjunto de atributos calculando:
  - La media de las correlaciones (o similar) de cada atributo con la clase
  - Las correlaciones por redundancias entre atributos

$$\text{Evaluación(Ai)} = \frac{\text{correlación con la clase}}{\text{correlaciones entre atributos}} = \sum_j U(A_j, C) \Big/ \sqrt{\sum_i \sum_j U(A_i, A_j)}$$

Evaluación de subconjuntos: Correlation Feature Selection (CFS)

- Ventaja: Método rápido
- Problemas: elimina atributos redundantes, pero como ranker, puede eliminar atributos que por si solos no están correlacionados con la clase, pero con otro atributo si que lo están (ej: "inteligencia artificial")

TAXONOMÍA DE MÉTODOS DE SELECCIÓN DE ATRIBUTOS

	Filter	Wrapper
Ranking (atributos individuales)	Information Gain, Información Mútua, Chi-square	
Subset selection	Correlation Feature Selection (CFS)	

RELIEF

Ranking (evaluación y ordenación de atributos de manera individual y eliminación de los menos valorados)

Subset selection (búsqueda del subconjunto de atributos más relevante)

- Métodos de búsqueda:
  - Greedy stepwise:
    - Hacia adelante
    - Hacia atrás
  - Mejor primero (best first)
  - Búsqueda genética

Evaluación de subconjuntos: Wrapper

- Los métodos *Wrapper* evalúan un subconjunto de atributos ejecutando un algoritmo de minería de datos (MD) concreto, sobre un conjunto de entrenamiento
- El valor del subconjunto es el porcentaje de aciertos obtenido con esos atributos
- Ventajas:
  - Obtienen subconjuntos de atributos adecuados para un algoritmo de MD concreto
  - evalúan a los atributos de los subconjuntos de manera realmente conjunta
- Desventajas:
  - son muy lentos (hay que ejecutar un algoritmo de aprendizaje muchas veces)
  - Pueden llevar a sobreaprendizaje

Métodos de búsqueda

- BestFirst: Mejor primero (lento)
- ExhaustiveSearch: Búsqueda exhaustiva (muy lento)
- GeneticSearch: Búsqueda genética (rápido)
- GreedyStepWise: Escalada (muy rápido):
  - Selección hacia delante
  - Selección hacia detrás
- RankSearch: Primero ordena los atributos y después construye el subconjunto de manera incremental, en dirección del mejor al peor, hasta que no merece la pena añadir nuevos atributos (rápido)

TAXONOMÍA DE MÉTODOS DE SELECCIÓN DE ATRIBUTOS

	Filter	Wrapper
Ranking (atributos individuales)	Information Gain, Información Mútua, Chi-square	
Subset selection	Correlation Feature Selection (CFS)	

RELIEF

Ranking (evaluación y ordenación de atributos de manera individual y eliminación de los menos valorados)

Subset selection (búsqueda del subconjunto de atributos más relevante)

- Métodos de búsqueda:
  - Greedy stepwise:
    - Hacia adelante
    - Hacia atrás
  - Mejor primero (best first)
  - Búsqueda genética

ALGORITMO RELIEF

- Es realmente un algoritmo de filter Ranking, pero tiene ventajas de subset selection y Wrapper: es capaz de detectar interacciones entre atributos, siendo muy rápido (pero no detecta atributos redundantes)
- Ventajas: detecta atributos relevantes e incluso aquellos que funcionan bien en grupos
- Desventajas: no detecta atributos redundantes

ALGORITMO RELIEF

- Atributos normalizados
- Repetir muchas veces:
  - Selecciona aleatoriamente una instancia (dato) x
    - Selecciona la instancia más cercana de la misma clase (hit) y la instancia más cercana de la otra clase (miss)
    - Incrementa el peso de aquellos atributos que tienen el mismo valor para la instancia hit y distinto valor para la instancia miss
      - $W_i <= W_i – (x_i – hit_i)^2 + (x_i - miss_i)^2$

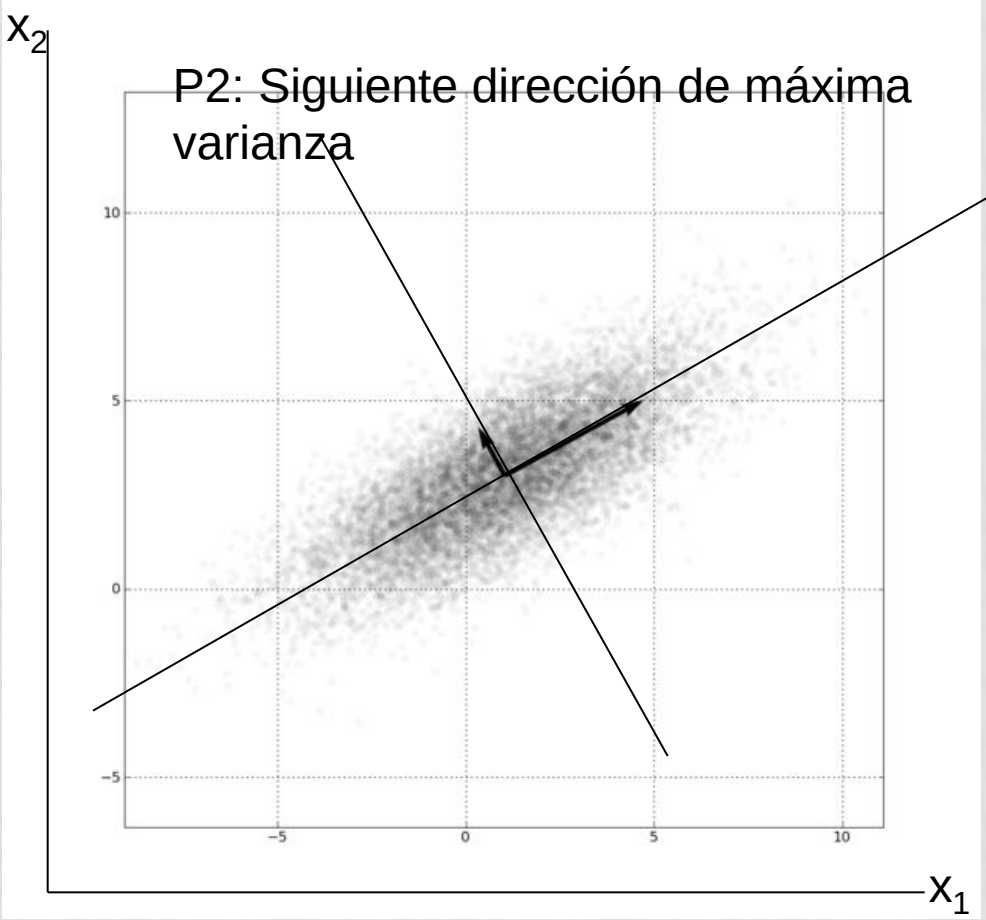
TRANSFORMACIÓN (+ SELECCIÓN) DE ATRIBUTOS

- Principal Component Analysis (PCA)
- Random Projections

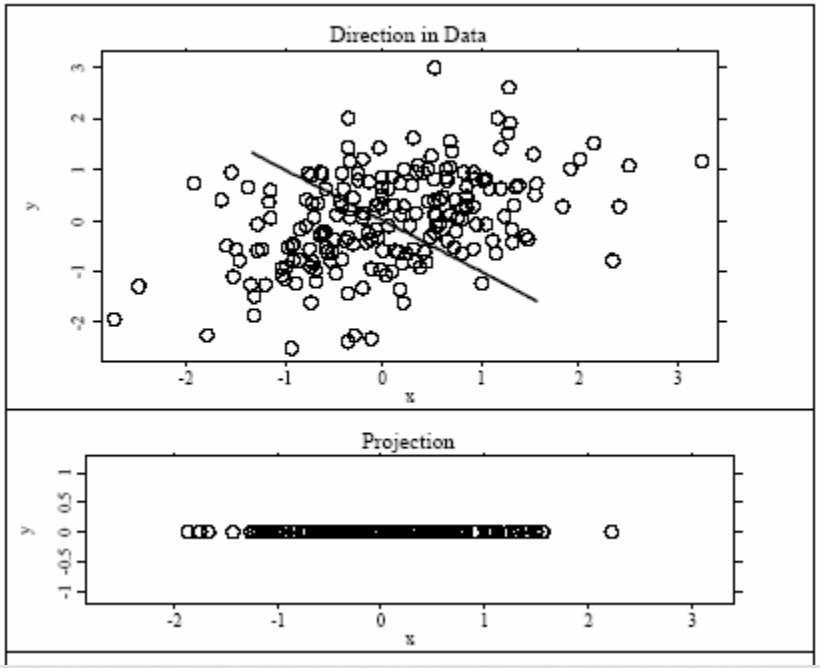
Selección con Principal Component Analysis (PCA)

- Este método construye nuevos atributos como combinación lineal de los anteriores
- Esos nuevos atributos están ordenados por importancia (varianza explicada)
- Se puede reducir la dimensionalidad escogiendo sólo algunos de los atributos

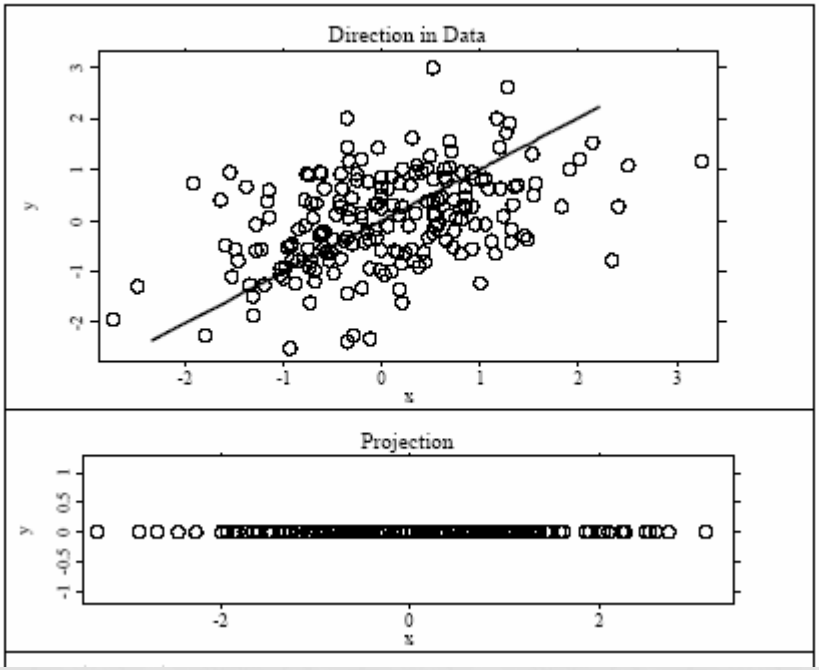
PCA



Crea dos atributos nuevos: P1 y P2

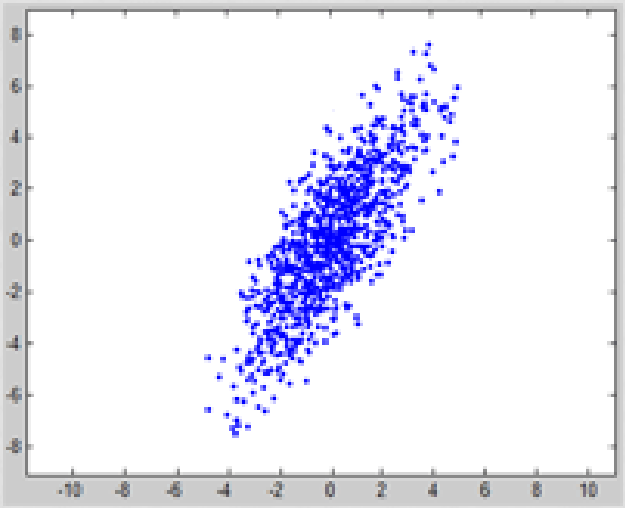

Direction in Data
y
x
Projection
y
x

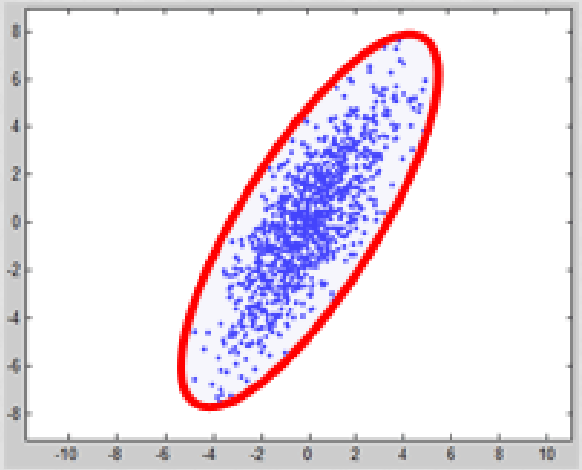
PEOR

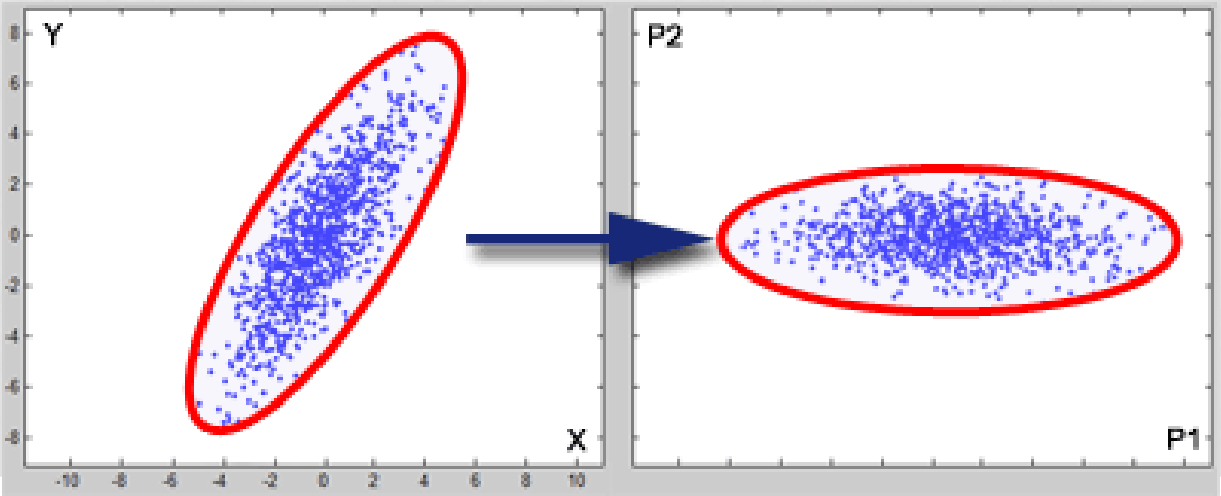

Direction in Data
y
x
Projection
y
x

MEJOR

Transformación realizada por PCA







- Se trata de transformaciones lineales
- Elimina redundancia (correlación) de los datos

$$P_1 = k_{11}*x_1+k_{12}*x_2$$

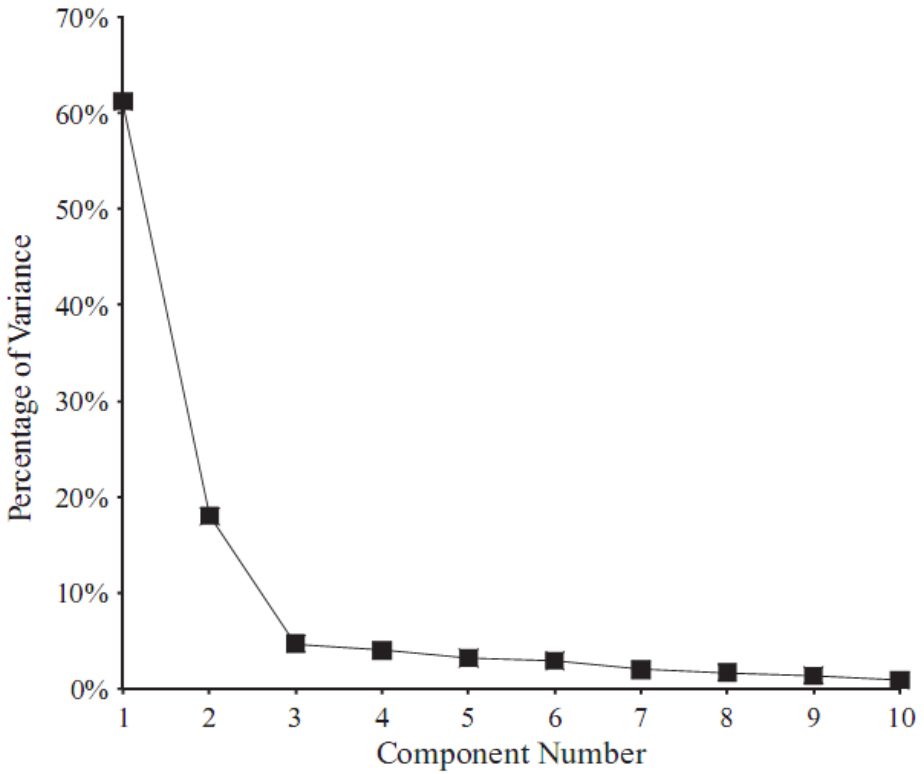
$$P_2 = k_{21}*x_1+k_{22}*x_2$$

$$P = \mathbf{X}*k$$

PCA: aparte de transformación, también selección

Axis	Variance	Cumulative
1	61.2%	61.2%
2	18.0%	79.2%
3	4.7%	83.9%
4	4.0%	87.9%
5	3.2%	91.1%
6	2.9%	94.0%
7	2.0%	96.0%
8	1.7%	97.7%
9	1.4%	99.1%
10	0.9%	100.0%

(a)



(b)

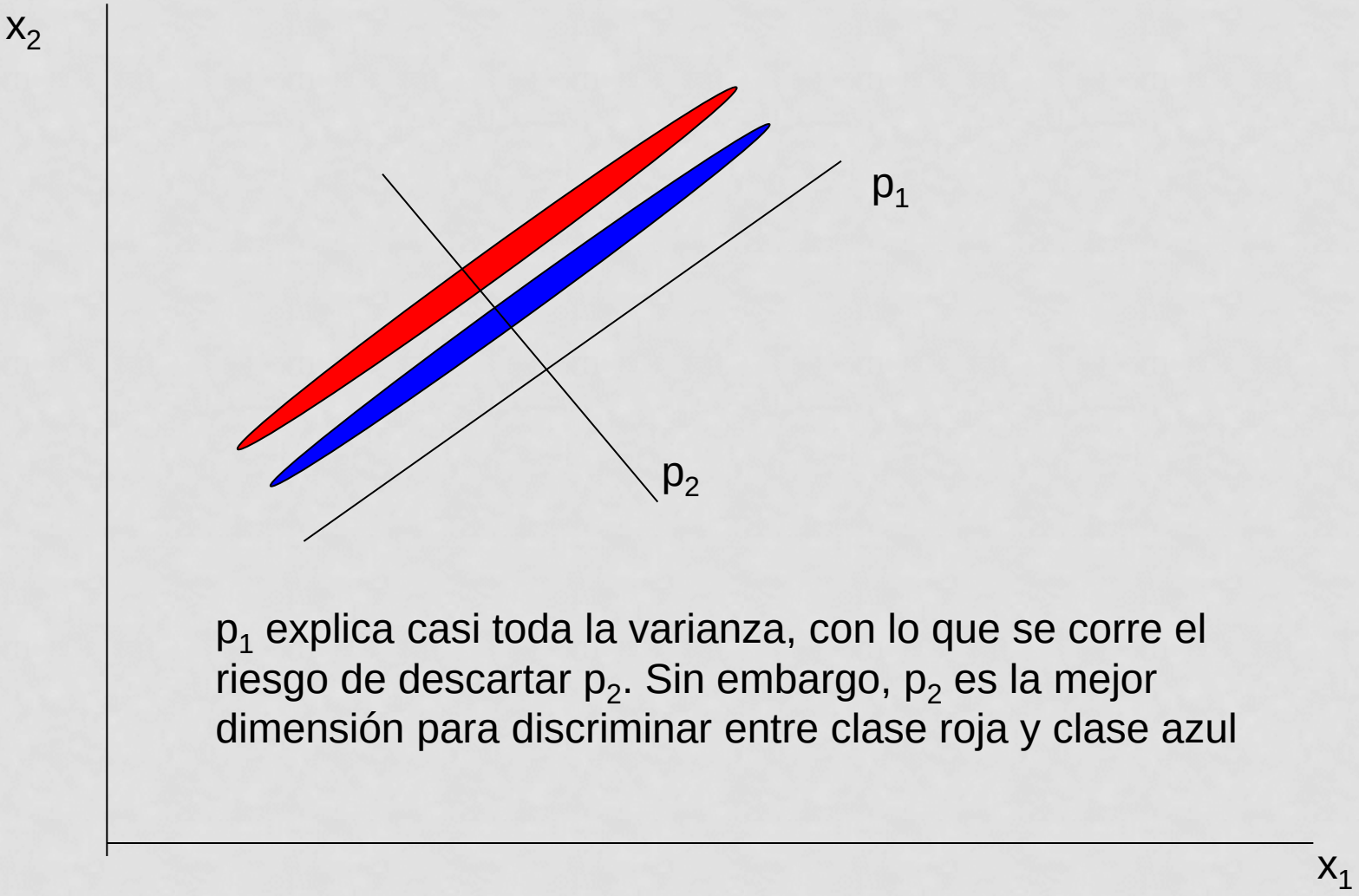
• Normalmente se suele coger tantos atributos como 95% de la varianza (7, en este caso)

•Si sólo unos pocos atributos explican la mayor parte de la varianza, el resto sobran (ej: datos en forma de elipse en 20 dimensiones)

Ventajas/desventajas de PCA

- Utilidad: puede determinar la dimensionalidad real de los datos
  - Ej: imaginar datos en forma de elipse de 2 dimensiones embebida en 20 dimensiones). PCA identificará fácilmente que con sólo 2 dimensiones se explica toda la varianza
- Utilidad: tiende a decorrelacionar los atributos
- Importante: PCA es un filtro no supervisado, con lo que no hay garantía de que genere los atributos que discriminan mejor las clases
- Desventaja: Si hay muchos atributos, es lento

Cuidado, PCA es no supervisado



p_1 explica casi toda la varianza, con lo que se corre el riesgo de descartar p_2. Sin embargo, p_2 es la mejor dimensión para discriminar entre clase roja y clase azul

Random projections

- Proyectar los datos a dimensiones inferiores mediante matrices aleatorias. Similares resultados a PCA y más rápido, siempre que el número de dimensiones de la proyección no sea demasiado pequeño
- X' = X*R
  - Dim(X) = num. datos x d
  - Dim(R) = dxd' ; d' << d
  - Dim(X') = num. Datos x d'
- Se puede demostrar que en X' se mantiene la estructura de los datos en X. Es decir, se mantienen aproximadamente las distancias entre datos

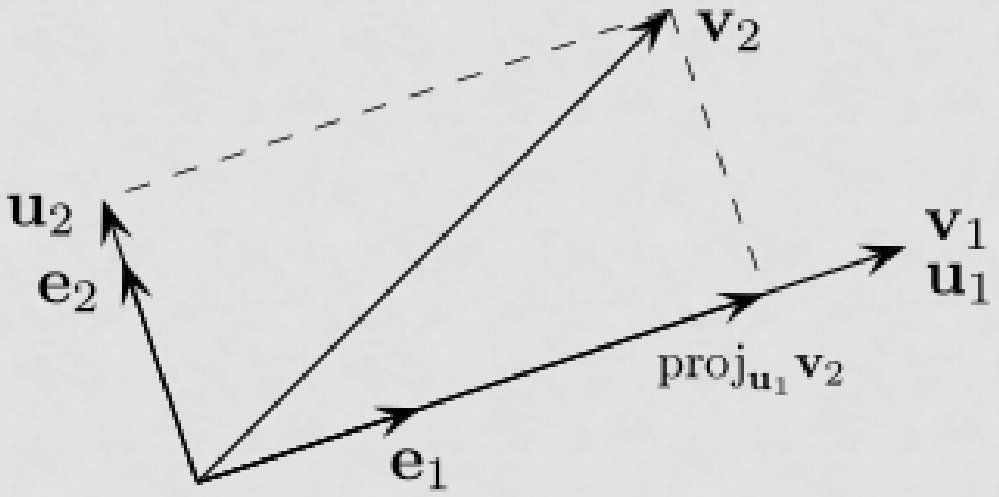
Random projections

- Pasos:
  1. Generar una matriz R con valores aleatorios en Normal(0,1)
  2. Ortogonalizar R (es decir, que las columnas de R sean vectores ortogonales), mediante por ejemplo, Gram-Schmidtt
  3. Normalizar las columnas de R al módulo unidad

- El paso 2 se puede obviar, porque en altas dimensiones, los vectores aleatorios son casi ortogonales

ORTOGONALIZACIÓN GRAM-SCHMIDT

- Se parte de v1,v2
- Se proyecta v2 sobre v1 = proj
- Se sabe que proj+u2 = v2, luego u2=v2-proj
- Normalizamos v1 y u2 y ya tenemos dos vectores ortogonales y unitarios e1 y e2



ORTOGONALIZACIÓN GRAM-SCHMIDT

$$\mathbf{u}_1 = \mathbf{v}_1,$$

$$\mathbf{u}_2 = \mathbf{v}_2 - \frac{\langle \mathbf{u}_1, \mathbf{v}_2 \rangle}{\langle \mathbf{u}_1, \mathbf{u}_1 \rangle} \mathbf{u}_1,$$

$$\mathbf{u}_3 = \mathbf{v}_3 - \frac{\langle \mathbf{u}_1, \mathbf{v}_3 \rangle}{\langle \mathbf{u}_1, \mathbf{u}_1 \rangle} \mathbf{u}_1 - \frac{\langle \mathbf{u}_2, \mathbf{v}_3 \rangle}{\langle \mathbf{u}_2, \mathbf{u}_2 \rangle} \mathbf{u}_2,$$

$$\mathbf{u}_k = \mathbf{v}_k - \sum_{j=1}^{k-1} \frac{\langle \mathbf{u}_j, \mathbf{v}_k \rangle}{\langle \mathbf{u}_j, \mathbf{u}_j \rangle} \mathbf{u}_j$$

Producto escalar:
u*v = |u| |v| cos(angulo) =
= |u| "proj v sobre u"

Random projections