

Minería de Datos

Preprocesamiento: Reducción de dimensionalidad

Dr. Edgar Acuña

Departamento de Ciencias Matemáticas
Universidad de Puerto Rico-Mayaguez

E-mail: edgar.acuna@upr.edu, eacunaf@gmail.com

Website: academic.uprm.edu/eacuna

Reducción de la Dimensionalidad

- **Selección de variables:** El principal objetivo de la selección de variables es reducir la dimensionalidad del espacio de variables, seleccionando variables relevantes y no redundantes. Una variable es redundante cuando da información contenida en alguna otra variable. Una variable es irrelevante si da muy poca información. Esto es, la selección de variables selecciona “q” variables del conjunto completo de “p” variables tal que $q \leq p$. Idealmente $q \ll p$.
- **Extracción de Variables:** Se construye un conjunto más pequeño de nuevas variables aplicando una transformación lineal (o no lineal) al conjunto original de variables. El método más conocido es el de Análisis de Componentes Principales (PCA). Otros: PLS, Curvas Principales.

Selección de Variables

Consideraremos solamente el problema de selección de variables para Clasificación Supervisada.

Objetivo: Seleccionar un subconjunto pequeño de variables tal que:

- a) La precisión del clasificador en el conjunto de datos no varíe de forma significativa.
- b) La distribución condicional resultante de una clase C , dado el vector de variables seleccionado G , esté tan cerca como sea posible a la distribución condicional original dadas todas las variables F .

Ventajas de la Selección de Variables

- El costo computacional de la clasificación será reducido ya que el número de variables será menor que antes.
- La complejidad del clasificador es reducida ya que se eliminan las variables redundantes y las irrelevantes.
- Ayuda a lidiar con el efecto de la “Maldición de la dimensionalidad”, que ocurre en procesos que requieren tener un gran número de instancias en comparación con el número de variables.

Pasos en la selección de variables

1. **Proceso de Generación:** La búsqueda del subconjunto óptimo puede ser: completo, heurístico, aleatorio.
2. **Función de Evaluación:** Medidas de distancia, medidas de información, medidas de consistencia, medidas de dependencia o de correlación, ChiSquare, tasa de error de clasificación.
3. **Criterio de parada:** Un umbral dado, número predefinido de iteraciones, tamaño predefinido del mejor subconjunto de variables.
4. **Proceso de Validación** (Opcional) Verificar si el subconjunto es válido.

Guía para escoger un método de selección de variables

- Habilidad para manejar distintos tipos de variables (continua, binaria, nominal, ordinal)
- Habilidad para manejar múltiples clases.
- Habilidad para manejar grandes conjuntos de datos.
- Habilidad para manejar datos ruidosos.
- Baja complejidad en tiempo.

Categorización de los métodos de selección de variables (Li, Chen y otros 2016)

<i>Medidas de Evaluación</i>	<i>Generación</i>		
	<i>Heurística</i>	<i>Completa</i>	<i>Random</i>
<i>Distancia</i>	Relief	Branch and Bound	-
<i>Información</i>	Trees	MDL	-
<i>Dependencia</i>	POEIACC	-	-
<i>Consistencia</i>	FINCO	Focus	LVF
<i>Tasa de Error de Clasificación</i>	SFS, SBS, SFFS	Beam Search	Genetic Algorithm

Los métodos de la ultima fila son también conocidos como los métodos “wrapper” (Kohavi, 1997). Los otros son llamados metodos de filtrado.

Métodos Híbridos

Son una combinación de los métodos de filtrado y “wrappers”. El ejemplo típico es el método de Máxima relevancia y Mínima Redundancia (Peng, Long and Ding, 2005)

Primero se determinan las variables más relevantes usando pruebas estadísticas de T o de F y luego se eliminan las redundantes usando medidas de información (entropía).

Métodos de Filtrado

Estos no requieren un clasificador, en lugar de ello usan medidas que permiten seleccionar las variables que distinguen mejor a la clase.

- RELIEF
- Las Vegas Filter (LVF)
- FINCO
- Otros : Branch & Bound, Focus,

El método RELIEF

- Kira and Rendell (1992) para problemas de dos clases y generalizado a problemas multi-clases por Kononenko (1994) y Kononenko, et al. (1997).
- Genera subconjuntos de variables en forma heurística.
- Una variable tiene un peso de relevancia, el cual es grande si puede distinguir claramente dos instancias que pertenecen a distintas clases pero no dos instancias que están en la misma clase.
- Usa una medida de distancia (Euclideana, Manhattan)

El método RELIEF (procedimiento)

- Se selecciona al azar un número dado ***Nsample*** de instancias del conjunto de entrenamiento D que contiene F variables.
- Los pesos de relevancia W_j de cada variable son inicializados en cero.
- Para cada instancia seleccionada \mathbf{x} , se debe identificar dos instancias particulares:

Nearhit: La instancia mas cercana a \mathbf{x} que pertenece a la misma clase.

Nearmiss: La instancia mas cercana a \mathbf{x} que pertenece a una clase diferente.

El método RELIEF (cont)

- Luego los pesos W_j 's ($i=1,..F$) son actualizados para cada instancia de la muestra tomada, usando la relación
$$W_j = W_j - \text{diff}(x_j, \text{Nearhit}_j)^2/NS + \text{diff}(x_j, \text{Nearmiss}_j)^2/NS$$

Si la variable X_k es nominal o binaria, entonces:

- $\text{diff}(x_{ik}, x_{jk}) = 1$ para $x_{ik} \neq x_{jk}$
 $= 0$ en caso contrario.

Si la variable X_k es continua u ordinal entonces:

- $\text{diff}(x_{ik}, x_{jk}) = (x_{ik} - x_{jk})/c_k$, donde $c_k = \text{range}(X_k)$

Decisión: Si $W_j \geq \tau$ (un valor preestablecido, usualmente un valor positivo cerca de cero) entonces se selecciona la variable f_j

Programas para RELIEF (cont)

- La librería dprep tiene una funcion relief . Tambien esta la librería CORElearn hace reliefF
- En Rapidminer se elige primero el operador Modelling luego attribute weighting y luego weights for relief.
- El modulo Scikit-learn de Python tiene algunos métodos de Feature selection, pero no tiene el ReliefF.
- En WEKA hay que usar Select Attributes luego como Attribute Evaluator: ReliefFAttributeEval y como Search method: Ranker.
- En dprep solo se usa un vecino mas cercano. Pero Rapidminer y Weka pueden usar mas de un vecino.

Conjunto de datos: Breast-Wisconsin

- 699 instancias, 9 variables y dos clases (benigno o maligno). 16 instancias han sido eliminadas por tener valores faltantes.
- 1. Clump Thickness 2. Uniformity of Cell Size, 3. Uniformity of Cell Shape, 4. Marginal Adhesion, 5. Single Epithelial Cell Size, 6. Bare Nuclei, 7. Bland Chromatin 8. Normal Nucleoli 9. Mitoses.
- Cada variable tiene valores en un rango de 0 a 10.

Ejemplo de Relief: Breastw

```
> relief(breastw,600,0)
```

Variables que aparecen en por lo menos la mitad de las repeticiones ordenadas por su peso de relevancia promedio:

	feature	frequency	weight
[1,]	6	10	0.10913169
[2,]	4	10	0.05246502
[3,]	1	10	0.04682305
[4,]	9	10	0.03171399
[5,]	2	10	0.02869547
[6,]	3	10	0.02566461
[7,]	5	10	0.02512963
[8,]	7	10	0.02096502
[9,]	8	10	0.01708025

Variables Seleccionadas

```
[1] 6 4 1 9 2 3 5 7 8
```

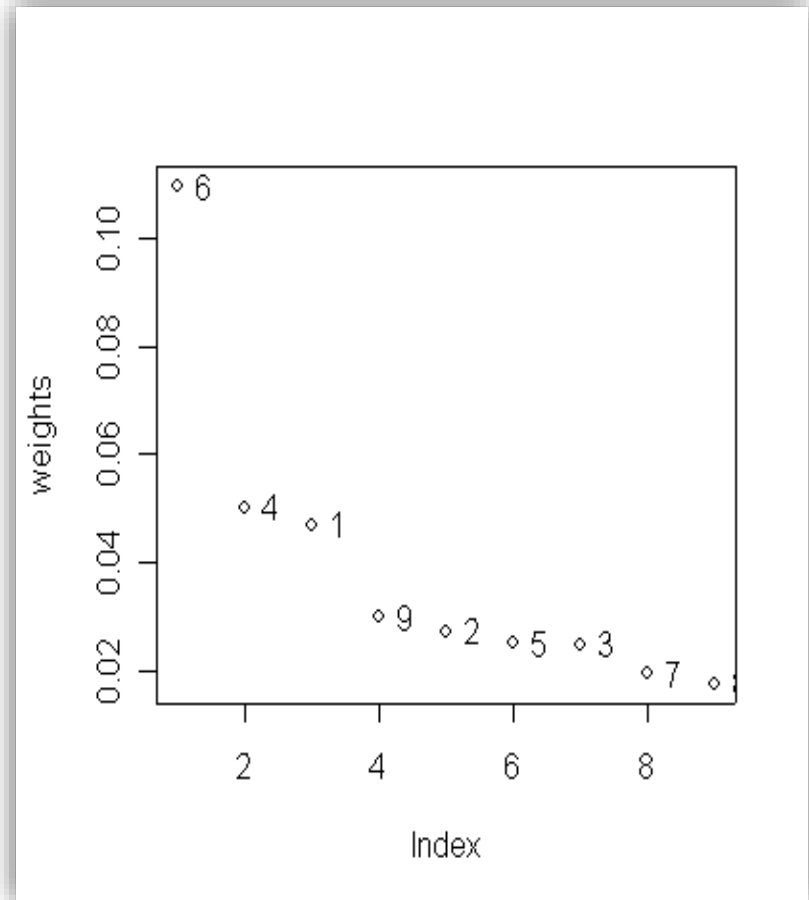
```
> relief(breastw,600,0.04)
```

Variables que aparecen en por lo menos la mitad de las repeticiones ordenadas por su peso de relevancia promedio:

	feature	frequency	weight
[1,]	6	10	0.10844239
[2,]	4	10	0.05293210
[3,]	1	10	0.04853909

Variables Seleccionadas

```
[1] 6 4 1
```



Método Relief: problema multi-clase

Primero se debe encontrar un **Nearmiss** por cada clase distinta a la que pertenece \mathbf{x} , y luego se promedia su contribución usando los pesos basados en anteriores. Los pesos son actualizados usando:

$$W_j = W_j - \text{diff}(x_j, \text{Nearhit})^2 + \sum_{C \neq \text{class}(x_j)} \frac{P(C)}{1 - P(\text{class}(x_j))} \text{diff}(x_j, \text{Nearmiss}(C))^2$$

Conjunto de datos: Landsat

- 6435 instancias, 36 predictoras continuas y 6 Clases
(suelo rojo, cultivo de algodón, suelo gris, suelo gris húmedo, suelo con vegetación rastroja, suelo gris bien húmedo)

```
> relief(Satellite,1500,0,1)
```

```
... features appearing in at least half of repetitions ordered by their average  
relevance weight:
```

	feature	frequency	weight
[1,]	25	1	0.12518576
[2,]	21	1	0.12121374
[3,]	13	1	0.12050074
[4,]	17	1	0.11886385
[5,]	33	1	0.11115147
[30,]	23	1	0.04898387

```
.....
```

[31,]	31	1	0.04776801
[32,]	12	1	0.04584572
[33,]	19	1	0.04535417
[34,]	11	1	0.04424004
[35,]	7	1	0.04287936
[36,]	8	1	0.04203942

```
selected features
```

```
[1] 25 21 13 17 33 29 9 5 1 26 14 22 18 30 2 34 10 36 3 16 27 6 20 28 4  
[26] 35 24 15 32 23 31 12 19 11 7 8
```

El método Relief (Cont)

Ventajas:

Trabaja bien con variables ruidosas y correlacionadas.
La complejidad en tiempo es lineal en el número de variables y en número de muestras tomadas (N_{sample}).
Trabaja para cualquier tipo de variable.

Desventajas:

Elimina las variables irrelevantes pero no elimina las variables redundantes.

- Elección del umbral.

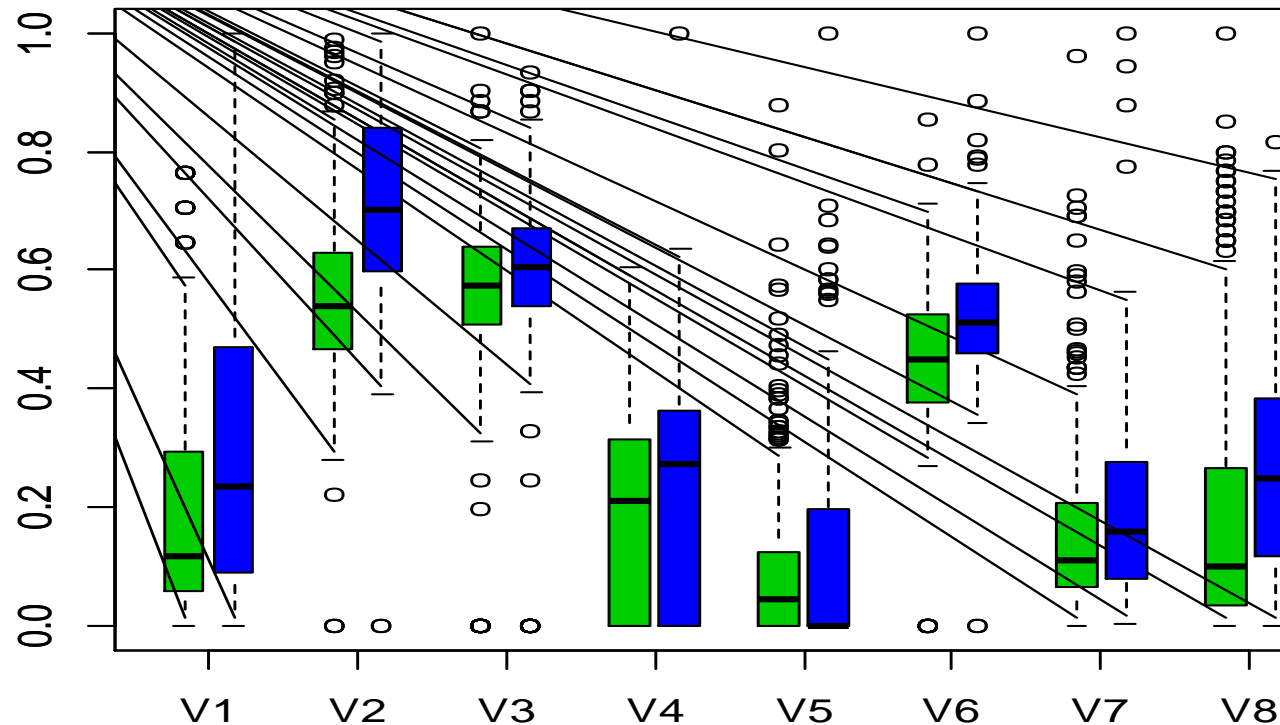
- Elección de N_{sample} .

Determinando visualmente las mejores variables

```
data(diabetes)
#normalizando diabetes
ndiab=mmnorm(diabetes)
a=ndiab[ndiab[,9]==1,]
b=ndiab[ndiab[,9]==2,]
#trazando boxplots comparativos
boxplot(a[, -9], at=0:7*3+1, main="comparando variables de diabetes", xaxt="n")
boxplot(b[, -9], at=0:7*3+2, xaxt="n", add=T)
axis(1, at = 0:7*3 + 1.5, labels = colnames(a)[-9], tick = TRUE)
```

Variables mas importantes en diabetes

Comparando variables por grupos



V2,V8, V1,V6 y V7 parecen ser las mas importantes

Usando pruebas estadísticas

En el caso de dos grupos se puede usar la prueba t y en el caso de mas de dos grupos la prueba F.

En el caso de atributos nominales o categoricos se prueba usar la prueba de Chi-square.

Todas esas pruebas estadísticas asumen propiedades distribucionales de lo datos que raras veces se cumplen, aun así se siguen usando.

Librería en Python para seleccion de variables

Disponible en <http://featureselection.asu.edu/>.

Tiene alrededor de 30 algoritmos para seleccion de variables, incluyendo Relief, F-score, Chi-Square, Ganancia de Informacion, mRMR y wrappers. Pero debe ser usado con precaucion por pequenos defectos.

Métodos Wrapper

Wrappers usan la tasa de error de mala clasificación, obtenida con un clasificador dado, como la función de evaluación para el subconjunto de variables.

- Selección secuencial hacia adelante (SFS)
- Selección secuencial hacia atrás (SBS)
- Selección secuencial flotante hacia adelante (SFFS)
- Otros: SFBS, Take I-remove r, Recursive Feature Elimination, GSFS, GA, SA.

Selección secuencial hacia adelante (SFS)

- Se inicializa el mejor subconjunto de variables T como el conjunto vacío.
- La primera variable que ingresa a T es aquella con la tasa de reconocimiento más alta con un clasificador dado.
- La segunda variable que ingresa a T será aquella que junto con la variable seleccionada en el paso previo produzca la tasa de reconocimiento más alta.
- El proceso continúa y en cada paso ingresa solamente una variable a T hasta que la tasa de reconocimiento no crezca cuando el clasificador se construya usando las variables que ya estén en T mas cada una de las demás variables.

Ejemplos: Bupa y Breastw

```
sfs(bupa,"knn") # clasificador knn con k=5 vecinos
```

El mejor subconjunto de variables es:

```
[1] 5 3 1
```

```
> sfs(bupa,"lda") #Linear discriminant classifier
```

El mejor subconjunto de variables es:

```
[1] 5 4 3 6
```

```
> sfs(bupa,"rpart") #decision tree classifier
```

El mejor subconjunto de variables es:

```
[1] 5 3 2
```

```
> sfs(breastw,"knn")#Knn con 5 vecinos
```

El mejor subconjunto de variables es:

```
[1] 6 1 3 7
```

```
> sfs(breastw,"lda")
```

El mejor subconjunto de variables es:

```
[1] 6 2 1 4
```

```
> sfs(breastw,"rpart")
```

El mejor subconjunto de variables es:

```
[1] 6 3 5
```

Selección secuencial hacia atrás (SBS)

- Inicialmente el mejor subconjunto de variables T incluyen todas las variables del conjunto de datos.
- En el primer paso, se realiza la clasificación sin considerar cada una de las variables, y se elimina la variable donde la tasa de reconocimiento es la más baja.
- El proceso continúa eliminando una variable en cada paso hasta que las tasas de reconocimiento con las variables usadas comienzan a decrecer.

No eficiente para los clasificadores no paramétricos porque tiene un alto tiempo de computación

Selección secuencial flotante hacia adelante(SFFS)

Pudil, et al (1994). Trata de resolver el problema de anidamiento que aparece en SFS y SBS.

- Inicialmente el mejor subconjunto de variables T es el conjunto vacío.
- En cada paso se incluye una nueva variable en T usando SFS, pero es seguida por una verificación de una posible exclusión de variables que ya estén en T. Las variables son excluidas usando SBS hasta que la tasa de reconocimiento comience a decrecer.
- El proceso continúa hasta que SFS no se pueda ejecutar.

Ejemplos

```
> sffs(bupa,"lda")
```

Las variables seleccionadas son:

```
[1] 3 4 5
```

```
> library(class)
```

```
> sffs(bupa,"knn") #knn con k=5 vecinos
```

Las variables seleccionadas son:

```
[1] 5 3
```

```
> library(rpart)
```

```
> sffs(bupa,"rpart")
```

Las variables seleccionadas son:

```
[1] 3 5 6 2
```

```
> sffs(breastw,"lda")
```

Las variables seleccionadas son:

```
[1] 1 2 6 4
```

```
> sffs(breastw,"knn")#lknn con 5 vecinos
```

Las variables seleccionadas son:

```
[1] 6 3 7 1
```

```
> sffs(breastw,"rpart")
```

Las variables seleccionadas son:

```
[1] 6 3 2
```

Conclusiones

- Entre los métodos wrapper, SFFS se comporta mejor que SFS: menor porcentaje de variables seleccionadas y casi la misma exactitud que SFS. Cómputo rápido.
- Entre los métodos de filtrado, FINCO tiene el menor porcentaje de variables seleccionadas.
- El rendimiento de LVF y RELIEF es bastante similar, pero LVF toma más tiempo de cómputo.
- Los métodos wrapper son más efectivos que los métodos de filtrado reduciendo el error de mala clasificación.
- La velocidad de cómputo de los métodos de filtrado es afectada por el tamaño de la muestra y el número de clases.

Conclusiones (Cont.)

- SFFS y FINCO tienen menor porcentaje de variables seleccionadas .
- En LVF, un incremento en el número de iteraciones disminuye la variabilidad de las variables seleccionadas.
- En LVF y FINCO, una reducción del nivel de inconsistencia mínimo aumenta el número de variables seleccionadas.

Extracción de Variables

Consideraremos solamente problemas de Clasificación Supervisada.

Objetivo: Construir nuevas variables a partir de las variables originales tal que la precisión del clasificador construido usando estas nuevas variables originales no varíe de forma significativa.

Veremos solamente el metodo de componentes principales.

Análisis de Componentes Principales (PCA)

El objetivo del Análisis de Componentes Principales (Hotelling, 1933) es reducir la información disponible. Esto es, la información contenida en p variables $\mathbf{X}=(X_1, \dots, X_p)$ puede ser reducida a $\mathbf{Z}=(Z_1, \dots, Z_q)$, con $q < p$. Las nuevas variables Z_i 's, son llamadas *Componentes Principales* y no son correlacionadas.

Los componentes principales de la matriz \mathbf{X} son los elementos de una transformación lineal ortogonal de \mathbf{X}

Desde un punto de vista geométrico, aplicar los componentes principales es equivalente a aplicar una rotación de los ejes coordenados.

Ejemplo: Bupa ($p=q=2$)

```
> bupapc=prcomp(bupa[,c(3,4)],scale=T,retx=T)
```

```
> print(bupapc)
```

Desviación estándar:

```
[1] 1.3189673 0.5102207
```

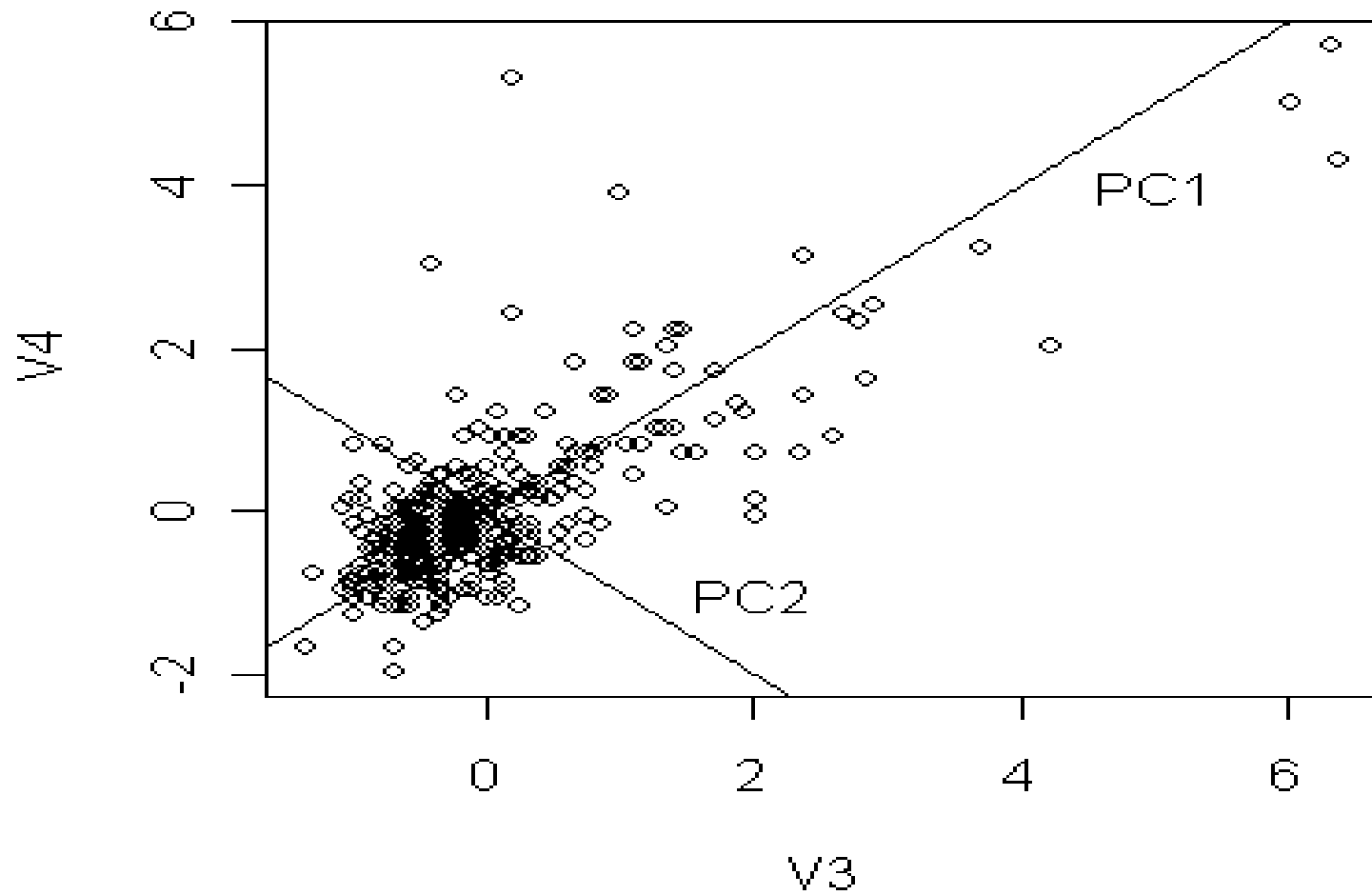
Rotation:

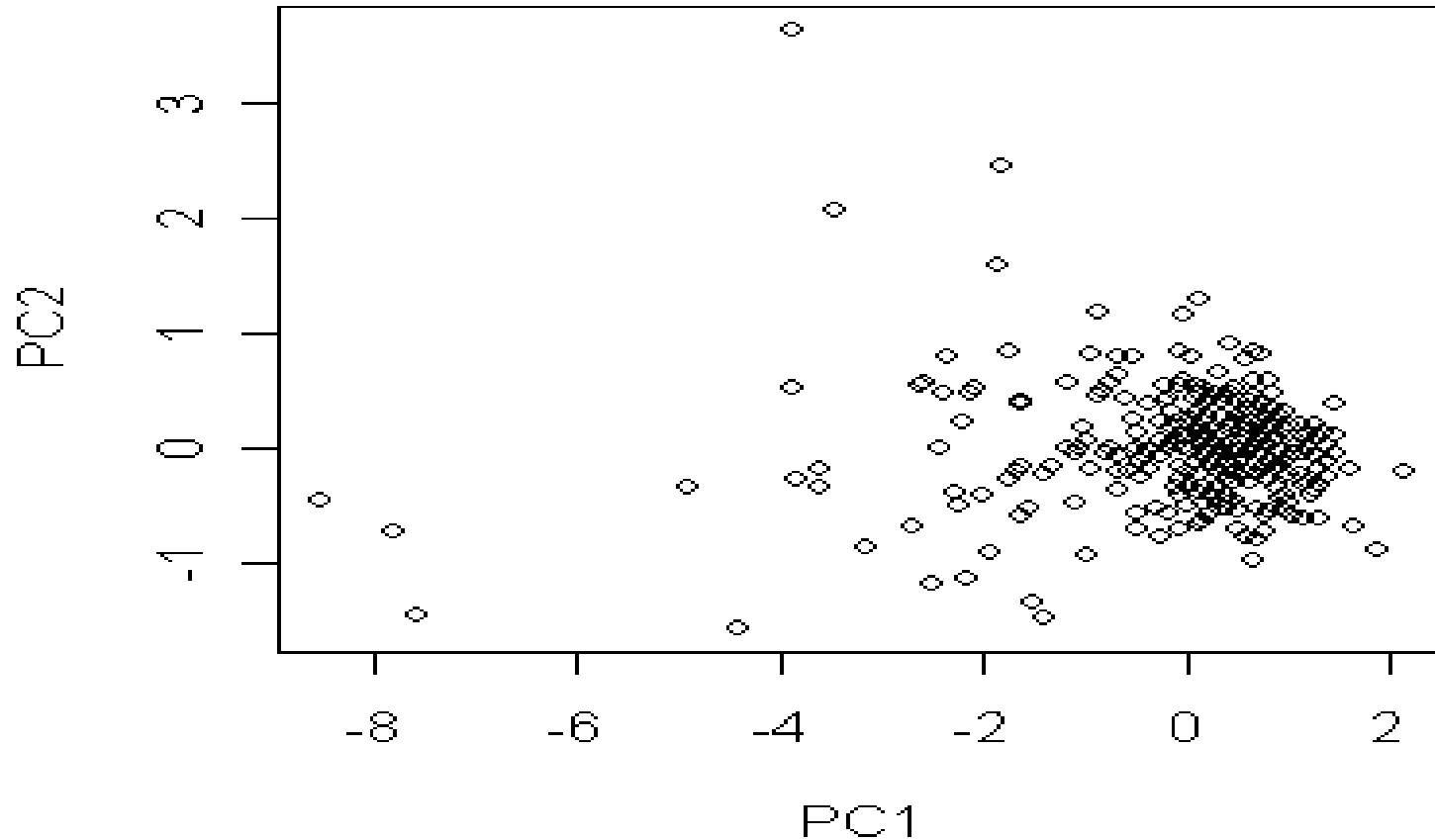
	PC1	PC2
--	-----	-----

V3	-0.7071068	-0.7071068
----	------------	------------

V4	-0.7071068	0.7071068
----	------------	-----------

effect of PCA





Notar que PC1 y PC2 no están correlacionadas

Hallando los Componentes Principales

Para determinar los Componentes Principales Z , debemos encontrar una matriz ortogonal V tal que

$$i) Z = \hat{X} V,$$

donde \hat{X} es obtenida de normalizar cada columna de X . Se puede demostrar que $VV' = V'V = I$

$$\text{y ii) } Z'Z = (\hat{X} V)'(\hat{X} V) = V'\hat{X}'\hat{X}V \\ = \text{diag}(\lambda_1, \dots, \lambda_p)$$

donde los λ_j 's son los valores propios de la matriz de correlación $\hat{X}'\hat{X}$. Así que V se encuentra usando la descomposición de valores de la matriz de correlación (o de covarianza).

La matriz V se llama matriz de cargas y contiene los coeficientes de todas las variables en cada componente principal.

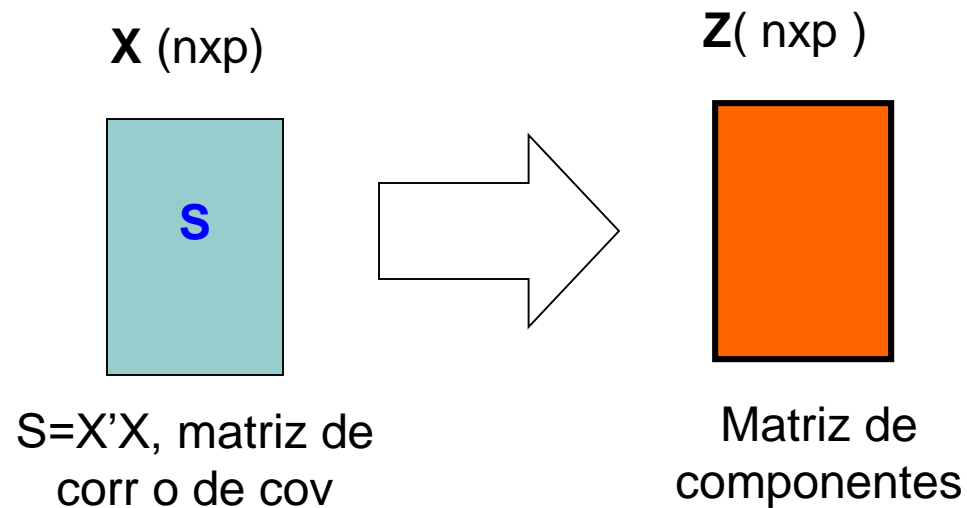
De (ii) el j-ésimo componente principal Z_j tiene una desviación estándar $\sqrt{\lambda_j}$ y puede ser escrito como:

$$Z_j = v_{1j}\hat{X}_1 + v_{2j}\hat{X}_2 + \dots + v_{pj}\hat{X}_p$$

Donde $v_{j1}, v_{j2}, \dots, v_{jp}$ son los elementos de la j-ésima columna en V y X_k es la k-esima columna de X .

Los valores calculados del componente principal Z_j son llamados los valores rotados o simplemente los “scores”.

PCA COMO UN PROBLEMA DE OPTIMIZACIÓN



$$\mathbf{Z}_k = \underset{\gamma' \gamma = 1}{\operatorname{argmax}} \operatorname{var}(\mathbf{X}\gamma)$$

Sujeto a la restricción de ortogonalidad

$$\gamma_j' \mathbf{S} \gamma_k = 0 \quad \forall \quad 1 \leq j < k$$

Selección del número de componentes principales

En la practica solo se usa un numero pequeño de componentes principales. Existen muchas alternativas (Ferre, 1995), pero las más usadas son:

- ❑ Escoger el número de componentes con una proporción acumulada de valores propios (i.e, varianza) de por lo menos 75 por ciento.
- ❑ Escoger hasta el componente cuyo valor propio es mayor que 1. Usar “Scree Plot”.

Ejemplo: Bupa

```
> a=prcomp(bupa[,-7],scale=T)
```

```
> print(a)
```

Desviación estándar:

```
[1] 1.5819918 1.0355225 0.9854934 0.8268822 0.7187226 0.5034896
```

Rotation:

	PC1	PC2	PC3	PC4	PC5	PC6
V1	0.2660076	0.67908900	0.17178567	-0.6619343	0.01440487	0.014254815
V2	0.1523198	0.07160045	-0.97609467	-0.1180965	-0.03508447	0.061102720
V3	0.5092169	-0.38370076	0.12276631	-0.1487163	-0.29177970	0.686402469
V4	0.5352429	-0.29688378	0.03978484	-0.1013274	-0.30464653	-0.721606152
V5	0.4900701	-0.05236669	0.02183660	0.1675108	0.85354943	0.002380586
V6	0.3465300	0.54369383	0.02444679	0.6981780	-0.30343047	0.064759576

Ejemplo (cont)

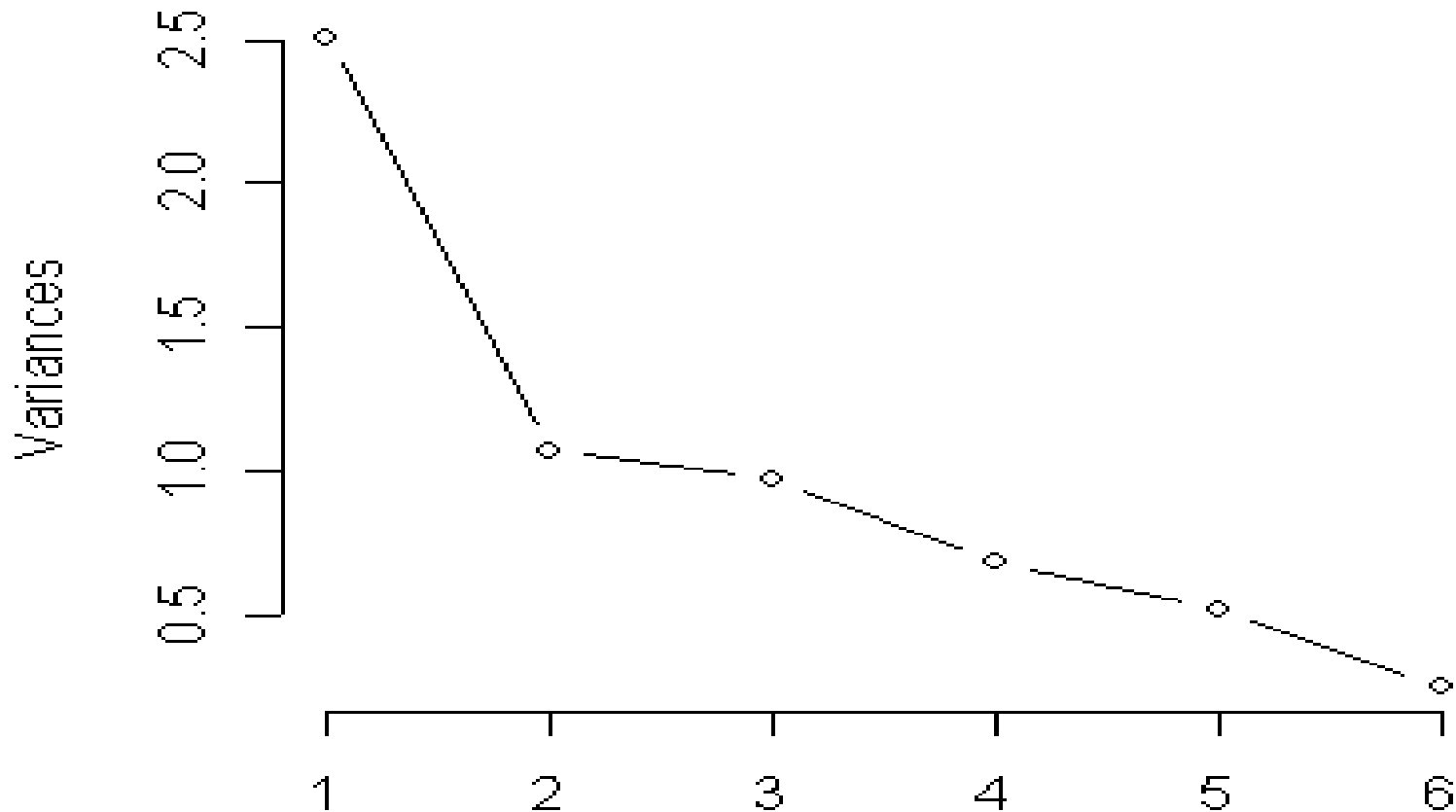
> summary(a)

Importancia de componentes:

	PC1	PC2	PC3	PC4	PC5	PC6
Desviación estándar	1.582	1.036	0.985	0.827	0.7187	0.5035
Proportion of Variance	0.417	0.179	0.162	0.114	0.0861	0.0423
Cumulative Proportion	0.417	0.596	0.758	0.872	0.9577	1.0000

>

screeplot of Bupa's PC



El screen-plot recomienda elegir dos componentes principales

PCA in Rapidminer

The screenshot displays the RapidMiner Studio Basic 6.5.001 interface. The main workspace shows a workflow titled "Main Process" with two operators: "Retrieve bupa" and "PCA (Principal Component Analysis)". The "Retrieve bupa" operator is connected to the "PCA" operator. The "PCA" operator has four output ports labeled "exa", "ori", "pre", and "res". The "exa" port is connected to the "PCA" operator's "exa" input port. The "ori" port is connected to the "PCA" operator's "ori" input port. The "pre" port is connected to the "PCA" operator's "pre" input port. The "res" port is connected to the "PCA" operator's "res" input port. The "PCA" operator is also connected to the "PCA" operator's "res" output port. The "PCA" operator's "exa" output port is connected to the "PCA" operator's "ori" input port. The "PCA" operator's "ori" output port is connected to the "PCA" operator's "pre" input port. The "PCA" operator's "pre" output port is connected to the "PCA" operator's "res" input port. The "PCA" operator's "res" output port is connected to the "PCA" operator's "exa" input port.

The left sidebar shows the "Operators" panel with a search bar containing "pri". Below it, the "Repositories" panel shows a list of data sources under "Local Repository (Edgar)", including "bupa" (Edgar - v1, 8/1/16 8:01 PM - 16 KB).

The right sidebar shows the "Parameters" panel for the "PCA (Principal Component Analysis)" operator. It includes a "dimensionality reduction" dropdown set to "keep variance" and a "variance threshold" input field set to "0.85".

At the bottom right, a "Help" panel titled "Principal Component Analysis (RapidMiner Studio Core)" provides a synopsis: "This operator performs a Principal Component Analysis (PCA) using the covariance matrix. The user can specify the..."

PCA en Python

La Libreria scikit-learn hace PCA
From sklearn.decomposition import PCA

Comentarios

- Muchos estudios han demostrado que PCA no da buenas predicciones en clasificación supervisada.
- Mejores alternativas: PLS (partial least squares) generalizados (Vega,2004) y PCA supervisados (Hastie, Tibshirani, 2004, Acuna and Porras, 2006).
- Para mejorar el rendimiento de componentes principales es mejor hacer primero selección de variables.