REPORTE DE INVESTIGACION

del Instituto de Cibernética, Matemática y Física

Métodos de extracción de características en datos de microarrays de ADN para enfermedades oncológicas (Parte II)

Miguel A. Gutiérrez Arce,
Yunier E. Tejeda Rodríguez,
Carlos Rodríguez Fadragas

Octubre 2017 ICIMAF 2017-839

ICIMAF

Calle 15 No 551 e/ C y D, Vedado, La Habana 4 C.P. 10400, Cuba. Telf.: (537) 32 7764, 32 2688.
Fax: (537) 33 3373 Télex: 512230 ICIMAF CU.

ISSN 0138-8916

Reporte de Investigación del ICIMAF No. 839 (2017)
ISSN 0138-8916 Original paper

Reporte de Investigación del ICIMAF
Calle 15 No. 551 e/ C y D, Vedado,
La Habana, Cuba, CP 10400,
http://www.icimaf.cu

Métodos de extracción de características en datos de microarrays de ADN para enfermedades oncológicas (Parte II).

[1]Miguel A. Gutiérrez Arce, [2]Yunier E. Tejeda Rodríguez, [3]Carlos Rodríguez Fadragas
[2]**Dpto Matemática UCLV**, [3]**Dpto Física UCLV**
[1]mgarce@uclv.cu, [2]yunier@uclv.cu, [3]fadragas@uclv.edu.cu

Resumen: En este reporte se realiza una caracterización de los algoritmos aleatorios, en particular la descomposición matricial CUR. Se proponen dos metodologías cuya idea central es la descomposición CUR para la obtención de un modelo de clasificación. Por último, se hace una discusión de los resultados por ambas metodologías.

Palabras claves: Algoritmos aleatorios, Descomposición matricial CUR, Descomposición del valor singular, Programación paralela, Reducción de la dimensión.

Abstract: In this report a characterization of the random algorithms, in particular the matrix decomposition CUR is performed. We propose two methodologies whose main idea is the decomposition of CUR to obtain a classification model. Finally, a discussion of the results is made by both methodologies.

Keywords: Random algorithms, CUR matrix decompositions, Singular value decomposition, Parallel programming, Dimension reduction.

1. Introducción.

En el Reporte I se realizó una caracterización de los métodos PCA, SPCA y PLS en cuanto a su fundamentación teórica, selección del número de componentes latentes y ordenamiento de las variables con respecto a dichas componentes. Además, se comentaron los principales problemas que presentan los datos de microarrays de ADN. Se presentó una metodología que consta de tres etapas para obtener un modelo de clasificación que pronostique diferentes enfermedades oncológicas. Por último, se hizo una discusión de los resultados que se obtuvieron.

A continuación, se presenta como está estructurado este reporte. En la sección 2 se caracterizan los algoritmos aleatorios en datos de microarrays de ADN, en particular la descomposición matricial CUR. En la siguiente sección se proponen dos metodologías que utilizan la descomposición matricial anterior para la obtención de un modelo de clasificación.

En la sección 4 se presentan los resultados y en la quinta se realiza la discusión de los mismos. Por último, se dan las conclusiones.

2. Algoritmos aleatorios en datos de microarrays de ADN.

Los algoritmos aleatorios para problemas matriciales muy grandes han recibido gran atención en los años recientes, principalmente en el análisis de microarrays de ADN. Estos algoritmos se refieren a una clase de algoritmos de proyección aleatoria y de muestreo aleatorio desarrollados recientemente [1]. Los algoritmos de proyección aleatoria resuelven el problema de aproximación de mínimos cuadrados mientras que los algoritmos de muestreo aleatorio resuelven el problema de aproximación de matrices de bajo rango [1]. Este último, es de nuestro interés en la investigación en el cual la descomposición matricial CUR [2] juega un papel muy importante.

2.1. Descomposición matricial CUR.

La descomposición matricial CUR se emplea para nombrar aquellas descomposiciones matriciales de bajo rango que son explícitamente expresadas en términos de un número pequeño de columnas y/o filas de una matriz de datos, A. Estas descomposiciones permiten aproximar la matriz A por medio del producto de tres matrices C, U y R donde C y R contienen algunas columnas y filas de A, respectivamente, mientras U es una matriz que se construye cuidadosamente de manera que garantice dicha aproximación.

Se conocen varias descomposiciones CUR que se diferencian en las cotas de error obtenidas y en el criterio para elegir las columnas y filas que forman las matrices C y R [3-7].

En [4] se propone la descomposición matricial CUR, la cual elige las columnas a incluir en C (similarmente en R) a partir de un factor de importancia para cada columna de la matriz A. Dicho factor se define a partir de la matriz A y un parámetro de entrada dado por el rango k, como se muestra a continuación:

$$\pi_j = \frac{1}{k}\sum_{p=1}^{k}(v_j^p)^2\,, \forall j = 1,\ldots,n \qquad (1)$$

donde v_j^p es la j-ésima componente del p-ésimo vector singular derecho de A.

A continuación, se muestrea aleatoriamente un pequeño número de columnas de A usando ese factor de importancia como una distribución de probabilidad.

El algoritmo básico para seleccionar las columnas de una matriz denominado ColumnSelect [4] toma como entrada cualquier matriz de orden $m \times n$, un parámetro de rango k y un parámetro de error ϵ.

El resultado teórico más importante que avala dicho algoritmo establece que con probabilidad al menos del 99%, esta elección de columnas satisface que

$$\|A - P_C A\|_F \leqslant \left(1 + \frac{\epsilon}{2}\right) \|A - A_k\|_F \qquad (2)$$

donde $P_C A$ denota la matriz de proyección sobre el espacio columna generado por C y A_k es la matriz de rango k más próxima a A en norma de Frobenius (ver en [8] la demostración).

De esta forma el resultado garantiza que si A es una matriz cercana a una matriz de rango k entonces, con alta probabilidad, el subespacio generado por las columnas de A está próximo al subespacio generado por las columnas de C. Esto justifica el hecho de poder utilizar un método en forma paralela distribuyendo la matriz de datos A en varias matrices C [9].

En [10] se mencionan los métodos experimentales "random", "exact.num.random", "top.scores", "ortho.top.scores" y "highest.ranks" los cuales se encuentran implementados en el paquete rCUR [11]. En dicho trabajo comentan que tales métodos proporcionan la misma precisión que el algoritmo ColumSelect.

3. Metodología propuesta

En esta sección se proponen dos metodologías que utilizan la descomposición matricial CUR para la obtención de un modelo de clasificación. La primera reduce la dimensión de los datos por CUR y luego emplea los métodos PCA, SPCA y PLS. Esta metodología es una continuación de la metodología presentada en el Reporte I. En lo adelante, cuando se refiera a esta metodología se hará por **Doble reducción de la dimensión**. La segunda es una realización en forma paralela de **Doble reducción de la dimensión**.

3.1. Doble reducción de la dimensión.

A continuación, se presenta la metodología que consiste de cinco etapas para la obtención un modelo de clasificación:

Etapa I: Reducción de la dimensión por CUR.

Etapa II: Reducción de la dimensión por métodos de extracción de características.

Etapa III: Construcción del modelo de clasificación.

Etapa IV: Validación del modelo de clasificación.

Etapa V: Ordenamiento de las variables respecto a la componente latente.

En la primera etapa se reduce la dimensión de los datos mediante la descomposición matricial CUR para obtener la matriz C.

A partir de la matriz C se reduce la dimensión empleando los métodos de extracción de características lineales siguientes:

- Análisis de Componentes Principales.
- Análisis de Componentes Principales Supervisados.
- Mínimos Cuadrados Parciales.

Luego de obtener las k-ésimas componentes mediante los métodos anteriores, se construye un modelo de clasificación por Análisis Discriminante Lineal (LDA, por sus siglas en inglés), utilizándose en dependencia del conjunto de datos los siguientes criterios:

- Para los conjuntos de microarrays que no presentan conjunto de prueba se utiliza validación cruzada dejando uno fuera.
- Para los conjuntos de microarrays que tienen conjunto de prueba se construye el modelo de clasificación con el conjunto de entrenamiento y se emplea validación "holdout" [12] para el conjunto de prueba.

Posteriormente se obtiene la matriz de confusión para cada método de extracción y se emplean las medidas de sensibilidad (Se), especificidad (Es) y exactitud (Ex) para determinar cuan bueno es el modelo de clasificación.

Estas, se describen en términos de positivos verdaderos (PV), negativos verdaderos (NV), negativos falsos (NF) y positivos falsos (PF):

$$Se = \frac{PV}{PV + NF}; 0 \leqslant Se \leqslant 1 \quad (3)$$

$$Es = \frac{NV}{NV + PF};\ 0 \leqslant Es \leqslant 1 \quad (4)$$

$$Ex = \frac{PV + NV}{PV + NV + NF + PF};\ 0 \leqslant Ex \leqslant 1 \quad (5)$$

La sensibilidad y especificidad son medidas que permiten indicar que el modelo de clasificación soluciona el problema del desbalance de las clases, mientras que la exactitud muestra que dicho modelo enmienda el problema de la complejidad de los datos.

En correspondencia al método de extracción de característica empleado, se ordenan las variables de acuerdo a su factor de importancia:

- PCA: $imp_j = \sum_{i=1}^{k} cor^2(x_j, u_i), \forall j = 1, \dots, p$
- SPCA: $imp_j = cor(x_j, u_{\theta,1}), \forall j = 1, \dots, p$
- PLS: $imp_j = -|\mathrm{w}_{j1}|, \forall j = 1, \dots, p$

3.2. Doble reducción de la dimensión en forma paralela.

La metodología **Doble reducción de la dimensión** propuesta en la sección anterior considera un modelo que tal vez no sea el más idóneo en la predicción. Con el propósito de eliminar esta problemática de selección del modelo correcto, se presenta una metodología cuya idea central es transformar el problema de clasificación de gran dimensión en varios problemas de clasificación de menores dimensiones, distribuyendo los datos en forma paralela con el objetivo de encontrar una cantidad $k < p$ de variables que tengan una alta capacidad predictiva. Dicha metodología referida por **Doble reducción de la dimensión en forma paralela** cuenta con seis etapas, realizándose las cuatro primeras en forma paralela. A continuación, se presenta dicha metodología:

Etapa I: Reducción de la dimensión por CUR.

Etapa II: Reducción de la dimensión por métodos de extracción de características.

Etapa III: Construcción de los m modelos de clasificación.

Etapa IV: Validación de los m modelos de clasificación.

Etapa V: Selección del mejor modelo.

Etapa VI: Ordenamiento de las variables respecto a la componente latente.

La primera etapa consiste en reducir la dimensión de los datos mediante la descomposición matricial CUR, obteniéndose m matrices C de modo que sus columnas satisfagan la desigualdad (2).

Una vez obtenidas las m matrices C se reduce la dimensión empleando los métodos de extracción de características lineales siguientes:

- Análisis de Componentes Principales.
- Análisis de Componentes Principales Supervisados.
- Mínimos Cuadrados Parciales.

Luego de obtener las k-ésimas componentes mediante los métodos anteriores, se construye un modelo de clasificación por LDA a cada matriz de datos reducidos, utilizándose en dependencia del conjunto de datos los siguientes criterios:

- Para los conjuntos de microarrays que no presentan conjunto de prueba se utiliza validación cruzada dejando uno fuera.
- Para los conjuntos de microarrays que tienen conjunto de prueba se construye el modelo de clasificación con el conjunto de entrenamiento y se emplea validación "holdout" [12] para el conjunto de prueba.

Posteriormente se obtiene la matriz de confusión para el método de extracción empleado en cada uno de los m modelos, y se utilizan las medidas de sensibilidad (Se), especificidad (Es) y exactitud (Ex) para determinar cuan bueno es el modelo de clasificación.

A partir de la validación de cada modelo se selecciona aquel modelo que tenga el menor error de mala clasificación, el cual viene definido como:

$$E = \min\{E_i\}_{i=1}^{m} = \min\{1 - Ex_i\}_{i=1}^{m} \qquad (6)$$

Luego, el modelo seleccionado es el modelo que presenta el mayor porciento de exactitud.

De acuerdo al método que se elija para la extracción de características, se ordenan las variables tomando en cuenta su factor de importancia:

- PCA: $imp_j = \sum_{i=1}^{k} cor^2(x_j, u_i), \forall j = 1, \dots, p$
- SPCA: $imp_j = cor(x_j, u_{\theta,1}), \forall j = 1, \dots, p$
- PLS: $imp_j = -|\mathrm{w}_{j1}|, \forall j = 1, \dots, p$

4. Resultados.

En el Reporte I se realizó una comparación entre los resultados obtenidos por los modelos PCA, SPCA y PLS, considerándose este último superior a los primeros en cuanto a los valores de sensibilidad, especificidad y exactitud. Por esta razón, en este reporte se implementan en el software R [13] las metodologías **Doble reducción de la dimensión** y **Doble reducción de la dimensión en forma paralela** utilizando el método PLS, denotadas como CUR-PLS y CUR-PLS-Par, respectivamente. En los **Anexos 1** y **2** se muestran el pseudocódigo para ambas implementaciones.

En la primera etapa que contempla la reducción de la dimensión por CUR, se utiliza el 10% del total de variables en cada conjunto de datos para determinar la matriz C. Con este fin, CUR-PLS selecciona aquellas variables con factores de importancia mayores por medio del método "top.scores". Para la ejecución del CUR-PLS-Par se crea un clúster de 4 procesadores a través de los paquetes foreach [14] y doSNOW [15] calculándose 100 matrices C en forma paralela mediante el método "random".

Una vez realizada la primera etapa, se procede a calcular k componentes latentes por PLS para reducir la dimensión. Para ello, se trabaja con una propuesta por validación cruzada de [16, 17] implementada en el paquete plsgenomics [18].

Luego de calcular las k componentes se pasa a la obtención un modelo de clasificación por LDA. Para lograr esto, CUR-PLS toma como variables predictoras estas k componentes obtenidas a partir de la matriz C. En cambio, CUR-PLS-Par lo hace de un modo diferente. A partir del clúster de 4 procesadores se calculan en forma paralela 100 modelos tomando como variables predictoras las k componentes obtenidas en las 100 matrices C. Seguido de esto, se realiza una validación cruzada deja-uno-fuera para cada modelo y se selecciona aquel que tenga el menor error de mala clasificación. Para la obtención de los modelos se utiliza el paquete MASS [19]. En la **Tabla 1** se muestra el número de componentes para los modelos PLS, CUR-PLS y CUR-PLS-Par. Además, se evidencia cómo se resuelve el problema de la reducción de la dimensión.

Tabla 1: Número de componentes latentes para PLS, CUR-PLS y CUR-PLS-Par

Métodos	Colon	DLCBL	CNS	Ovarian	GLI85	SMK
PLS	5	1	1	5	4	2
CUR-PLS	1	1	1	7	2	1
CUR-PLS-Par	6	1	3	6	4	2

Por último, se valida el modelo de clasificación obtenido por CUR-PLS, así como CUR-PLS-Par empleando las medidas de sensibilidad (Se), especificidad (Es) y exactitud (Ex) para determinar cuan bueno es el modelo de clasificación. En la **Tabla 2** y la **Figura 1** se muestran dichas medidas. En los **Anexos 3** y **4** se encuentran las matrices de confusión para los modelos CUR-PLS y CUR-PLS-Par, respectivamente.

Tabla 2: Resultados para el LDA en los conjuntos de datos.

Métodos	Medidas	Colon	DLCBL	CNS	Ovarian	GLI85	SMK
PLS	Ex	**0.92**	**1**	0.72	0.99	0.98	0.73
	Se	**0.95**	**1**	0.97	0.99	**1**	0.80
	Es	**0.86**	**1**	0.24	0.99	0.92	0.64
CUR-PLS	Ex	0.82	0.91	0.77	0.98	0.89	0.74
	Se	0.88	0.92	0.87	0.99	0.98	0.80
	Es	0.73	0.91	0.57	0.97	0.69	0.67
CUR-PLS-Par	Ex	**0.92**	**1**	**0.93**	**1**	**0.99**	**0.75**
	Se	**0.95**	**1**	**0.92**	**1**	**1**	**0.82**
	Es	**0.86**	**1**	**0.95**	**1**	**0.96**	**0.68**

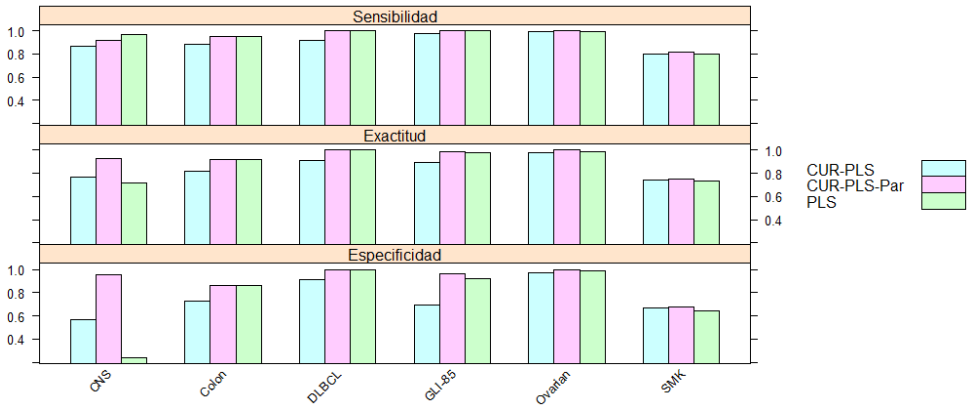

Figura 1: Resultados para el LDA en los conjuntos de datos.

5. Discusión.

En la **Tabla 2** se puede apreciarse que el modelo CUR-PLS no presenta valores en sensibilidad, especificidad y exactitud por encima de los exhibidos por PLS, salvo en determinados conjuntos como son CNS y SMK. Esto demuestra que la metodología **Doble reducción de la dimensión** no es del todo eficiente en la predicción de enfermedades aun cuando el modelo es más simple debido a que el número de variables representa el 10% de las originales.

En el modelo CUR-PLS-Par se observan mejores resultados que el CUR-PLS, dado que en los conjuntos de datos que se estudian, las medidas empleadas para validar dicho modelo son superiores. Por su parte, el modelo PLS da resultados muy similares, incluso iguales en determinados conjuntos al modelo CUR-PLS-Par. La diferencia entre ambos modelos radica en el número de variables originales a reducir, pues PLS trabaja con todas las variables mientras que CUR-PLS-Par lo hace con el 10% de ellas.

A modo de conclusión se puede decir que el modelo CUR-PLS-Par es el mejor resolviendo los problemas de reducción de la dimensión, desbalance y solapamiento de las clases. Además, la programación paralela brinda un aporte sustancial en este resultado, pues se selecciona de 100 modelos el de mayor exactitud.

6. Conclusiones.

- Se caracterizan los algoritmos aleatorios en la aplicación de los datos de microarray de ADN en particular la descomposición CUR.
- Se proponen las metodologías **Doble reducción de la dimensión** y **Doble reducción de la dimensión en forma paralela** para la obtención de un modelo de clasificación.
- Se implementaron las metodologías mencionadas anteriormente en el entorno de desarrollo integrado RStudio para que pueda ser empleada en el software R.
- Los resultados evidencian como el modelo CUR-PLS-Par resuelve los problemas de reducción de la dimensión, desbalance y solapamiento de las clases.
- La programación paralela brinda un aporte sustancial en la predicción de muestras cancerígenas y no cancerígenas

RECONOCIMIENTOS

Los autores desean agradecer las contribuciones que ha hecho el MSc. Jorge Luis Morales Martínez y a todas aquellas personas que han contribuido de alguna manera para que sea posible la realización del presente trabajo. Para todos ellos, MUCHAS GRACIAS…"

Referencia

1. Mahoney, M.W., *Randomized algorithms for matrices and data*. 2001, Stanford University. p. 1-54.
2. Martinsson, P.-G., *Randomized methods for matrix computations and analysis of high dimensional data*. 2016. p. 1-55.
3. A. Frieze, R.K.a.S.V., *Fast Monte-Carlo algorithms for finding low-rank approximations.* J. ACM,, 2004. **51**: p. 1025–1041.

4. Drineas, M.W.M.a.P., *CUR matrix decompositions for improved data analysis.* PNAS. **106**: p. 697-702.
5. P. Drineas, R.K.a.M.W.M., *Fast Monte-Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition.* SIAM J Comput, 2006. **36**: p. 184–206.
6. Stewart, G.W., *Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix.* Numer. Math, 1999. **83**: p. 313-323.
7. Tyrtyshnikov, S.A.G.a.E.E., *The maximum-volume concept in approximation by low-rank matrices.* Contemporary Mathematics, 2001. **280**: p. 47–51.
8. P. Drineas, M.W.M.a.S.M., *Relative-error CUR matrix decompositions.* SIAM J Matrix Anal Appl, 2008. **30**: p. 844-881.
9. V. Bolón-Canedo, N.S.-M.a.A.A.-B., *Distributed feature selection: An application to microarray data classification.* Applied Soft Computing, 2015. **30**: p. 136-150.
10. A. Bodor, I.C., M. W. Mahoney and N. Solymosi *rCUR: an R package for CUR matrix decomposition.* BMC Bioinformatics, 2012. **13** p. 1-6.
11. Solymosi, A.B.a.N. *rCUR: CUR decomposition package*. 2012; Available from: http://CRAN.R-project.org/package=rCUR.
12. Bolón-Canedo, V., *Novel feature selection methods for high dimensional data*, in *Department of Computer Science*. 2014, UNIVERSITY OF A CORUÑA.
13. Team, R.C., *R: A Language and Environment for Statistical Computing*, R.F.f.S. Computing, Editor. 2016: Vienna, Austria.
14. Weston, R.A.a.S. *foreach: Foreach looping construct for R*. 2014; Available from: https://CRAN.R-project.org/package=foreach.
15. Weston, R.A.a.S. *doSNOW: Foreach parallel adaptor for the snow package*. 2014; Available from: https://CRAN.R-project.org/package=doSNOW.
16. Boulesteix, A.-L., *PLS Dimension Reduction for Classification with Microarray Data.* Statistical Applications in Genetics and Molecular Biology, 2004. **3**(1): p. 1-32.
17. Boulesteix, A.-L.I., *Dimension Reduction and Classication with High-Dimensional Microarray Data*, in *F akultät für Mathematik, Informatik und Statistik*. 2004, Ludwig-Maximilian-Universitä at München. p. 1-116.
18. Anne-Laure Boulesteix, G.D., Sophie Lambert-Lacroix, Julie Peyre and Korbinian Strimmer. *plsgenomics: PLS Analyses for Genomics*. 2015; Available from: https://CRAN.R-project.org/package=plsgenomics.
19. Ripley, W.N.V.a.B.D., *Modern Applied Statistics with S*. Fourth ed. 2002, New York: Springer.

Anexo

Anexo 1: Pseudocódigo para la propuesta **Doble reducción de la dimensión**.

Datos a entrar:

X: Datos de microarrays.

Y: Variable dependiente binaria. La clase "1" representa que el paciente no padece la enfermedad. La clase "2" representa que el paciente si padece la enfermedad.

type: Una variable que toma los valores "PCA" y "SPCA".

Hacer:

type = "PCA"

C = CUR(X)

if (type = "PCA") then

  Y_K = PCA(C)

else if (type = "SPCA") then

  Y_K = SPCA(Y, C)

else

  Y_K = PLS(Y, C)

end if

$\hat{Y}$ = LDA (Y, Y_K)

mc = matriz.confusion (Y, $\hat{Y}$)

Ex = sum(diag(mc))/sum(mc)

Se = mc [2,2]/sum(mc [2,])

Es = mc [1,1]/sum(mc [1,])

Comentarios:

C: Matriz obtenida por el algoritmo COLUMNSELECT aplicado a la matriz X.

PCA: Análisis de Componentes Principales.

SPCA: Análisis de Componentes Principales Supervisados.

PLS: Mínimos Cuadrados Parciales.

Y_K: K primeras componentes latentes.

LDA: Análisis Discriminante Lineal.

$\hat{Y}$: Estimación de la variable dependiente.

mc: Matriz de confusión de orden 2x2.

Ex: Exactitud.

Se: Sensibilidad.

Es: Especificidad.

Anexo 2: Pseudocódigo para la propuesta **Doble reducción de la dimensión en forma paralela**.

Datos a entrar:

X: Datos de microarrays.

Y: Variable dependiente binaria. La clase "1" representa que el paciente no padece la enfermedad. La clase "2" representa que el paciente si padece la enfermedad.

type: Una variable que toma los valores "PCA" y "SPCA".

Hacer:

type = "PCA"

for ($i = 1$ to m) do

  C_i= CUR(X)

  if (type = "PCA") then

    Y_{k_i} = PCA (C_i)

  else if (type = "SPCA") then

    Y_{k_i} = SPCA(Y, C_i)

  else

    Y_{k_i} = PLS(Y,C_i)

  end if

  $\hat{\mathrm{Y}}_i$ = LDA (Y, Y_{k_i})

  mc_i = matriz.confusion (Y, $\hat{\mathrm{Y}}_i$)

  Ex_i = sum(diag(mc_i))/sum(mc_i)

  Se_i = mc_i [2,2]/sum(mc_i [2,])

  Es_i = mc_i [1,1]/sum(mc_i [1,])

end for

best = min($1 - \mathrm{Ex}_1$, $1 - \mathrm{Ex}_2$, …, $1 - \mathrm{Ex}_m$)

$\hat{\mathrm{Y}}_{best}$

Comentarios:

C_i: Matriz i-ésima obtenida por el algoritmo COLUMNSELECT aplicado a la matriz X.

PCA: Análisis de Componentes Principales.

SPCA: Análisis de Componentes Principales Supervisados.

PLS: Mínimos Cuadrados Parciales.

Y_{k_i}: K primeras componentes latentes de la matriz C_i.

LDA: Análisis Discriminante Lineal.

$\hat{\mathrm{Y}}_i$: Estimación i-ésima de la variable dependiente.

mc_i: Matriz i-ésima de confusión cuyo orden es 2x2.

Ex_i: Exactitud del i-ésimo modelo.

Se_i: Sensibilidad del i-ésimo modelo.

Es_i: Especificidad i-ésimo modelo.

best: Menor error de mala clasificación.

$\hat{\mathrm{Y}}_{best}$: Mejor modelo de clasificación.

Anexo 3: Matriz de confusión para el modelo CUR-PLS.

Colon	FALSO	TRUE
FALSO	16	6
TRUE	5	35

DLBCL	FALSO	TRUE
FALSO	21	2
TRUE	2	22

CNS	FALSO	TRUE
FALSO	12	9
TRUE	5	34

Ovarian	FALSO	TRUE
FALSO	88	3
TRUE	2	160

GLI-85	FALSO	TRUE
FALSO	18	8
TRUE	1	58

SMK	FALSO	TRUE
FALSO	60	30
TRUE	19	78

Anexo 4: Matriz de confusión para el modelo CUR-PLS-Par.

Colon	FALSO	TRUE
FALSO	19	3
TRUE	2	38

DLBCL	FALSO	TRUE
FALSO	23	0
TRUE	0	24

CNS	FALSO	TRUE
FALSO	20	1
TRUE	3	36

Ovarian	FALSO	TRUE
FALSO	91	0
TRUE	0	162

GLI-85	FALSO	TRUE
FALSO	25	1
TRUE	0	59

SMK	FALSO	TRUE
FALSO	61	29
TRUE	17	80