Capítulo 2

Selección usando PCA y RNA's

2.1. Introducción

En esta sección se explica el proceso de selección de características utilizando Análisis de Componentes Principales (PCA) y Redes Neuronales Artificiales (RNA).

La sección 2.2, *Marco Teórico*, introduce al lector a los conceptos necesarios para entender el uso PCA como técnica de selección de características, así como algunos conceptos necesarios para entender los ajustes que sufrió la entrada en el caso del HPV. La sección 2.3, *PCA para la selección de características*, explica el uso de PCA como técnica de selección de características paso a paso. La sección 2.4, *Caso de trabajo: HPV*, explica la forma en que se atacó el caso específico del HPV. La sección 2.5, *Resultados*, muestra los resultados obtenidos. La sección 2.6, *Conclusiones*, muestra las conclusiones referentes únicamente a la selección de características usando PCA y RNA's.

2.2. Marco Teórico

Para entender la forma en que es posible usar el análisis de componentes principales para la selección de características, primero debemos explicar las propiedades de éste.

La siguientes secciones abordan esta necesidad para después explicar el proceso de selección de características usando PCA.

2.2.1. Análisis de Componentes Principales

También es conocido como la transformada Karhunen-Loève [3]. El Análisis de Componentes Principales (PCA del inglés Principal Components Analysis) es una de las técnicas de análisis multivariante más sencillas [20]. Su objetivo es el de, dado un conjunto de variables $Y_1, Y_2, Y_3, \ldots, Y_p$, encontrar combinaciones lineales que produzcan nuevas variables $Z_1, Z_2, Z_3, \ldots, Z_p$, llamadas componentes principales, tal que éstas no estén correlacionados y la varianza esté maximizada en cada una de ellas. La falta de correlación entre los índices indica que éstos miden dimensiones diferentes de los datos. Además, los nuevos índices están ordenados en relación a su varianza.

La entrada al PCA consiste en los datos que desean analizarse en forma de una matriz, que llamaremos $INPUT$. Ésta tiene dimensiones $N \times M$, dónde cada N renglón corresponde a una muestra diferente y cada M columna corresponde a una características, variable o medición diferente. Si las características M miden datos de naturaleza diferente (distancia, velocidad, volumen, etc) es necesario hacer una normalización de los datos para que la diferencia en las escalas de valores no afecten la efectividad del algoritmo.

El resultado principal, producto de aplicar PCA sobre la matriz $INPUT$, es una matriz $TRANS$ de tamaño $M \times M$. Dicha matriz contiene los los M polinomios lineales que permiten calcular los M componentes principales buscados, como se muestra a continuación:

$$pc_{ij} = INPUT_i \cdot TRANS_j \tag{2.1}$$

donde $INPUT_i$ es el vector de variables para la muestra i, $TRANS_j$ es el vector de la

matriz de transformación para el cálculo del j-ésimo componente principal y pc_{ij} es el j-ésimo componente principal para la i-ésima muestra.

Como ya dijimos, los componentes principales se calculan y ordenan de tal manera que:

$$v(pc_1) > v(pc_2) > \ldots > v(pc_M) \tag{2.2}$$

donde $v(x)$ es la varianza de x y pc_M es el m-ésimo componente principal. Nótese que nos referimos a $v(pc_M)$ pues nos referimos a la varianza del componente M con respecto a las N muestras.

Además la varianza total de los datos originales se mantienen, es decir:

$$\sum_{m=1}^{M} v(pc_m) = \sum_{m=1}^{M} v(INPUT_m) \tag{2.3}$$

2.2.2. Compresión de datos

Como mencionamos en la sección 1.2.1, para el caso de la selección con PCA y RNAs, es necesario comprimir los datos sobre los fragmentos de restricción a un solo valor. Para esto exploraremos dos técnicas, la primera el uso de PCA como un algoritmo de *reducción de la dimensionalidad* o *compresión*, y la segunda el uso de redes neuronales en forma de *cuello de botella*.

Compresión usando PCA

Cuando hay una alta correlación entre las variables originales [20], el análisis de componentes principales resulta de mucha utilidad en aplicaciones de reconocimiento de patrones. En estos casos es común que la desigualdad mencionada en la ecuación 2.2 sea muy pronunciada, al grado en que muchos de los componentes principales generados contengan una varianza mínima o nula. En muchos casos, ésto nos permite representar el total de las variables originales utilizando solamente una fracción de los componentes

principales generados. Debido a esta propiedad, el PCA se ha convertido en un procedimiento común para reducir la dimensionalidad de los datos de entrada en una gran variedad de aplicaciones.

Para ilustrar un poco mejor la capacidad de reducción de la dimensionalidad podemos agregar como ejemplo general que, si tuviéramos una matriz como $INPUT$ donde los M características que medimos de las N muestras están altamente correlacionadas, es probable que después de generados los M componentes principales un clasificador solamente necesitara de un número menor al 10 % del total de los M componentes principales para identificar correctamente N muestras originales. Así pues, en lugar de usar las M variables originales, utilizamos un número reducido de los componentes principales generados incorrelacionados, lo que en muchos casos aumenta la efectividad del clasificador al mismo tiempo que reduce sus necesidades de procesamiento y memoria.

Compresión usando RNA-Cuello de botella

Las redes neuronales artificiales han sido usadas extensivamente para la compresión de datos [28]. En el caso del presente trabajo utilizaremos RNA's con una topología de cuello de botella pues es fácil observar la afinidad de dicho enfoque con los objetivos de este paso. Además este tipo de redes pueden implementarse de manera sencilla con el software que tenemos a nuestro alcance.

Las *RNA's de cuello de botella* trabajan como codificadores-decodificadores. Son redes alimentadas hacia delante (FF por su nombre en inglés, Feed-Forward) cuya topología es de la forma (i, h, i) donde $h < i$. Se pueden entrenar con cualquier algoritmo para el entrenamiento de redes FF aunque generalmente se utiliza retropropagación o alguna de sus variantes.

El objetivo del entrenamiento es que la salida de la red reproduzca la entrada así que durante el entrenamiento los vectores de entrada (*Input*) son iguales a los vectores de

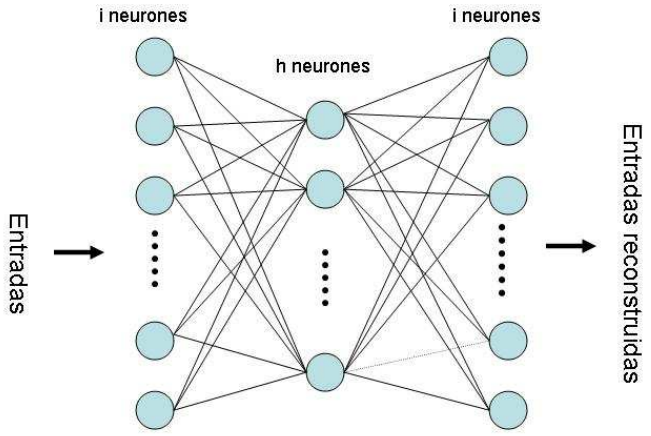

Figura 2.1: Topología de una RNA cuello de botella

salida deseada (*Target*). Si al final de entrenamiento se logra cumplir este objetivo tendremos que la capa intermedia de h neurones estará actuando como un codificador y la capa de salida de i neurones como un decodificador, como podemos ver en la figura 2.1. Una vez entrenada la red se separan la capa de entrada y la capa intermedia de la capa de salida, y se utilizan las primeras dos como un codificador que arroja h valores.

2.3. PCA para la selección de características

Aunque no es su función principal, en [20], [7], [17] se menciona el uso de esta técnica para evaluar la importancia de las variables en un conjunto de datos, de lo que se intuye su capacidad como técnica de selección. En [21], y [5] se hace referencia al PCA explícitamente como técnica de selección de características.

La capacidad del PCA para funcionar como técnica de selección de características puede notarse si observamos la matriz de transformación $TRANS$, recordamos la propiedad descrita en la ecuación 2.2 y el significado de la varianza.

La varianza de una variable x en una colección de datos es un número no negativo que nos indica que tanto se aleja la población del valor promedio de dicha variable.

En el caso del reconocimiento de patrones, una varianza elevada en cierta variable puede indicarnos que dicha variable tiene mayor capacidad para separar las clases del problema.

Ya mencionamos que en muchos casos la varianza total del sistema puede ser contenida en unos cuantos componentes principales. El cálculo de cada componente principal se obtiene de la forma:

$$pc_m = TRANS_{m1}INPUT_{i1} + TRANS_{m2}INPUT_{i2} + \ldots + TRANS_{mp}INPUT_{im} \quad (2.4)$$

donde pc_m es el m-ésimo componente principal y Y_p es la p-ésima característica o variable.

Los coeficientes $TRANS_{mp}$ de cada uno de estos polinomios pueden servir como un indicador de la *importancia* que tiene cierta variable en el cálculo de cierto componente principal. Es natural suponer que aquellas características o variables p cuyos coeficientes $TRANS_{mp}$ sean elevados para los primeros componentes principales, serán aquellas características que aporten la mayor capacidad de separación. Las características que se escojan de esta manera serán los elementos de ϕ.

Es importante aclarar que la selección de dichas características sigue teniendo cierto grado de subjetividad [17]. Pues aunque se tienen métricas para comparar la varianza de los diferentes componentes principales y los polinomios para el cálculo de estos nos aportan una guia, no existe una metodología formal para realizar esta selección.

2.3.1. Descripción del proceso

Para formalizar el proceso y poder garantizar límites para nuestro método proponemos una metodología que sigue la forma de un algoritmo del tipo *hill climbing*. Nos

apoyamos con un clasificador, del cual observaremos su eficacia para clasificar las muestras. En el caso del presente trabajo, este clasificador será una RNA alimentada hacia adelante (RNAFF) entrenada por retropropagación.

La idea general del método es:

- Aplicar PCA sobre nuestra colección de datos.
- Buscar el conjunto $MINPCS$ de componentes principales mínimo necesario para poder clasificar correctamente todas las clases de nuestra colección de datos. Debido a que los PCS están ordenados en forma progresiva con respecto a su varianza, esta es una búsqueda particularmente sencilla.
- Una por una, agregar a nuestro conjunto de características ϕ (ver 1.2) aquellas que más participan en el cálculo de éstos componentes y verificar la efectividad del clasificador. Detenerse cuando el clasificador haya alcanzado el desempeño deseado.

Además, debido a que los componentes seleccionados se calculan con una combinación lineal de los datos originales podemos decir con seguridad que:

$$\text{si } M(MINPCS) = p \text{ entonces} \tag{2.5}$$

$$\exists \phi \in \Phi \text{ tal que } M(\phi) = p \tag{2.6}$$

donde $M(X)$ es una métrica de desempeño y Φ es el conjunto total de características seleccionables (ver 1.2). En el peor de los casos $\phi = \Phi$.

2.3.2. El proceso paso a paso

La figura 2.2 muestra los pasos para el proceso de selección de características propuesto. El primer paso es procesar la matriz $INPUT$ con el Análisis de Componentes

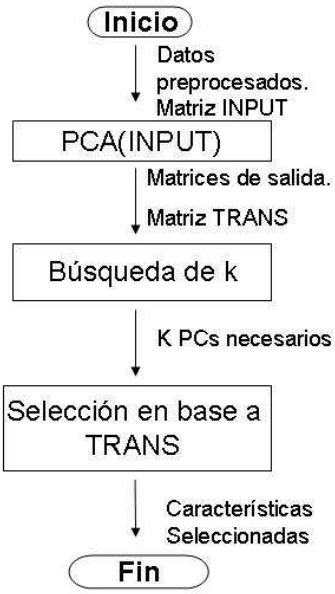

Figura 2.2: Diagramas de flujo del proceso de selección

Principales. Este proceso produce 4 matrices resultantes que enumeramos a continuación:

- TRANS. Matriz de dimensiones $M \times M$ que contiene los M polinomios que permiten el cálculo de los M componentes principales generados por el algoritmo PCA. Cada renglón contiene M elementos. Estos elementos son los coeficientes del polinomio que calcula el PC m.

- NEWDATA. Matriz de dimensiones $N \times M$ que contiene el valor de cada unos de los M PCS generados para las N muestras de la matriz $INPUT$. La matriz NEWDATA resulta de hacer $INPUT \times TRANS$.

- VARIANCE. Matriz de dimensiones $N \times 1$ que contiene la varianza contenida en cada componente principal.

- PERCENT_VARIANCE. Matriz de dimensiones $N \times 1$ que contiene el porcentaje

de varianza de cada componente principal en relación a la varianza total de los datos de $NEWDATA$.

El siguiente paso es buscar el número mínimo k de componentes necesarios para clasificar correctamente las clases del problema. La búsqueda se hace progresivamente, es decir primero se prueba usando la información del 1er PC, si no se alcanza el desempeño deseado agrega el 2o PC y así sucesivamente hasta encontrar un número de PC's adecuado. Recordemos que estamos usando RNA's como clasificadores, así pues en cada iteración también debemos buscar una topología de RNA que permita el desempeño deseado, solo si ésta no es encontrada procedemos a agregar un PC más.

El flujo de este paso se muestra en la figura 2.3. El proceso contiene 2 ciclos. El primer ciclo busca el número k de PC's de manera incremental. Se busca que k sea mínimo. El segundo ciclo busca una red que permita el reconocimiento de las clases del problema. La búsqueda del segundo ciclo es la búsqueda del número de neurones l en el nivel intermedio de una red con topología (k, l, i), las variables k, i están fijas en este ciclo pues: estamos validando el valor de k y el valor de i es siempre el número de clases que deseamos clasificar. Hay que mencionar que la búsqueda de i es un proceso lento por lo que es necesario limitarla. Los limites de esta búsqueda deben ser escogidos según la naturaleza del problema que se este tratando.

En el siguiente paso tomamos, de la matriz $TRANS$, los polinomios que permiten el cálculo los k componentes principales seleccionados en el paso anterior. Agregaremos características a ϕ una a una verificando la efectividad de ϕ con cada adición. El procedimiento para la adición de características a ϕ es el que se muestra en la figura 2.4.

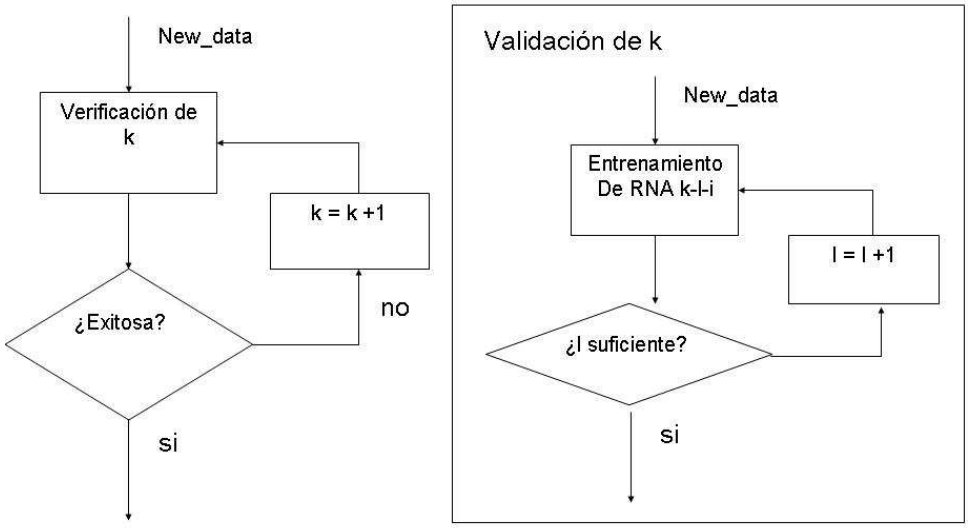

Figura 2.3: Diagramas de flujo del proceso de búsqueda de k.

1 for i = 1 to k

2 buscar característica m tal que $|TRANS_{im}| = max(|TRANS_{im}|_{m=1}^{M})$

3 $\phi = \{\phi, m\}$

4 if $M(\phi)$ >=eficacia requerida

5 break;

6 end(for)

Figura 2.4: Creación de ϕ usando RNA. $M(x)$ es una métrica de eficacía del conjunto de características x.

2.4. Caso de trabajo, HPV

El caso sobre el que probaremos este procedimiento, como dijimos en el capítulo 1, es la optimización de genotipificación del Papilomavirus Humano (HPV). Vimos anteriormente que este problema se traduce a la selección de enzimas. Las enzimas generan entre 1 y 4 salidas, las distancia de los fragmentos de restricción, por lo que nuestro procedimiento no se puede aplicar directamente.

Como vimos anteriormente, nuestro procedimiento requiere que la entrada tenga una forma tal que cada columna de ésta se relacione directamente con una característica. Por estas razones buscamos comprimir la información fragmentaria en un solo dato que representara adecuadamente la salida de las enzimas. El resultado sería que para cada enzima tendríamos una transformación que comprimiría su salida, que originalmente es de 4 valores, a un solo valor.

2.4.1. Compresión de fragmentos

Para comprimir la información fragmentaria probamos dos técnicas, compresión usando Redes Neuronales Artificiales y compresión usando Análisis de Componentes Principales. En cualquiera de los dos enfoques el objetivo es comprimir la información distribuida en 4 fragmentos a un solo valor.

Se trabaja con la compresión de los datos por enzima, es decir se analiza la posibilidad de compresión tomando cada enzima como un caso individual.

Como medida de efectividad, en los dos casos, utilizamos el porcentaje de reconocimiento obtenido de una RNA con una topología 1-48 con una función de activación *tangente-sigmoide*, después de haber sido entrenada por 50 iteraciones utilizando la implementación del algoritmo de entrenamiento *Levenberg-Marquardt* del paquete de RNA's de MatLab. Esta red se entrena para producir un vector de tamaño $1 \times i$, re-

cordamos que i es el número de clases del problema, en este caso $i = 48$. Cada posición corresponde a una clase y la red debe arrojar 1 en la posición de la clase a la que corresponda la muestra y 0's en las demás posiciones. Debido a que la red no arroja exactamente 0's o 1's por utilizar tangente-sigmoide como función de activación, utilizamos la distancia euclidiana para buscar el vector más cercano a la salida. Para medir la efectividad de la compresión contamos el número de fallos y aciertos de una red como la descrita anteriormente recién entrenada con los datos comprimidos.

Compresión usando RNA's

El procedimiento que seguiremos será exactamente el descrito en la sección 2.2.2.

Debido a que trabajamos con cada enzima de manera independiente tendremos 205 RNA's cuello de botella. Cada una de estas redes tiene la topología $(4, 1, 4)$ pues cada enzima produce 4 datos, la distancia recorrida por los fragmentos generados, y deseamos comprimir estos 4 datos en un solo valor, por esta razón tenemos 1 neurón en la capa intermedia. Después de verificar que la capa de salida produzca los valores deseados ésta es desechada.

La figura 2.5 muestra el proceso de creación de cada red.

La matriz m_{enz} tendrá 205 columnas donde cada columna k corresponde a las salidas de tamaño 1 de la red de cuello de botella para la correspondiente enzima k.

Compresión usando PCA

Aplicaremos PCA sobre los datos generados por las enzimas de manera independiente. Así pues, tendremos 205 matrices de tamaño 48, una para cada enzima. El resultado de cada uno de estos análisis son 4 matrices con las características de las matrices que describimos en la sección 2.3.1 y con dimensiones $j = 4$ y $i = 48$.

Figura 2.5: Proceso de creación de una RNA cuello de botella

Esperamos poder observar en la mayor parte de los casos, las matrices *PERCENT_VARIANCE* que el primer componente principal contenga casi toda la varianza de los datos originales, indicando que éste es representativo de los 4 valores originales. También se valida que sea posible obtener el mismo porcentaje de reconocimiento obtenido utilizando los 4 valores originales pero utilizando solamente el primer componente principal. Si estas dos condiciones se cumple podemos estar seguros que el el primer componente principal representa adecuadamente los datos originales y podemos utilizarlo en lugar de éstos.

La figura 2.6 muestra el flujo del proceso de compresión usando PCA.

2.5. Resultados

Esta sección contiene los resultados de aplicar el proceso descrito en la sección 2.3.1 para el caso del HPV y los resultados de los procesos de compresión mencionados en la sección 2.4.1.

Figura 2.6: Proceso de compresión usando PCA

Empezamos por los resultados de la sección 2.4 pues estos resultados son necesarios para entender el resto.

2.5.1. Resultados de la compresión usando RNA's en forma de cuello de botella

Como se explicó en la sección 2.2.2, se crearon 205 RNA's en este paso. Se evaluaron 2 índices de efectividad. El primero es el error cuadrático medio obtenido durante el entrenamiento y el segundo el porcentaje de reconocimiento obtenido utilizando el nuevo valor comprimido.

Entrenamos por 100, 200, y 300 épocas y pudimos observar que los resultados se mantenían constantes. Esto indica que son necearías menos de 100 épocas para obtener los resultados deseados. La figura 2.7 muestra los ECM obtenidos.

Se obtuvo un ECM promedio = 0.7678, una mediana = 0.0505 con una desviación estándar = 1.3549. Valores satisfactorios para nuestros propósitos.

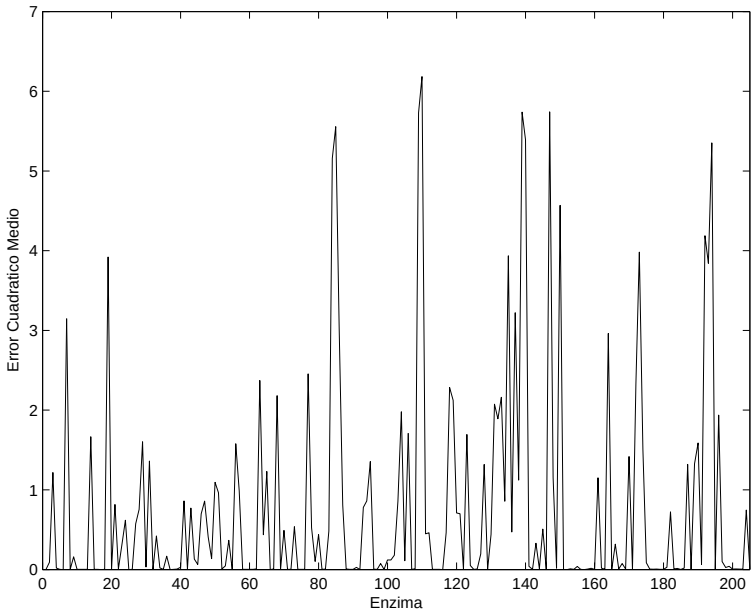

Figura 2.7: Errores cuadráticos medios obtenidos para la compresión de las 205 enzimas

El porcentaje de reconocimiento se mantuvo constante excepto para los casos de 5 enzimas. Dichas enzimas se caracterizan por generar alrededor de 3 fragmentos en promedio.

2.5.2. Resultados de la compresión usando PCA

En este caso analizamos 2 puntos, la varianza contenida en el primer componente principal y el porcentaje de reconocimiento obtenido usando como entrada usando éste valor.

La varianza del 1er componente mostró una mediana = 98.6771, una media = 93.9450 y una desviación estándar = 9.3634 cuando se aplicó PCA. Esto nos indica que en la mayoría de los casos el primer componente principal logró retener la mayor parte de la varianza total de los datos originales.

El porcentaje de reconocimiento se comportó de la misma manera que en el caso de compresión usando RNA's. Esta vez pudimos observar que en los casos donde el

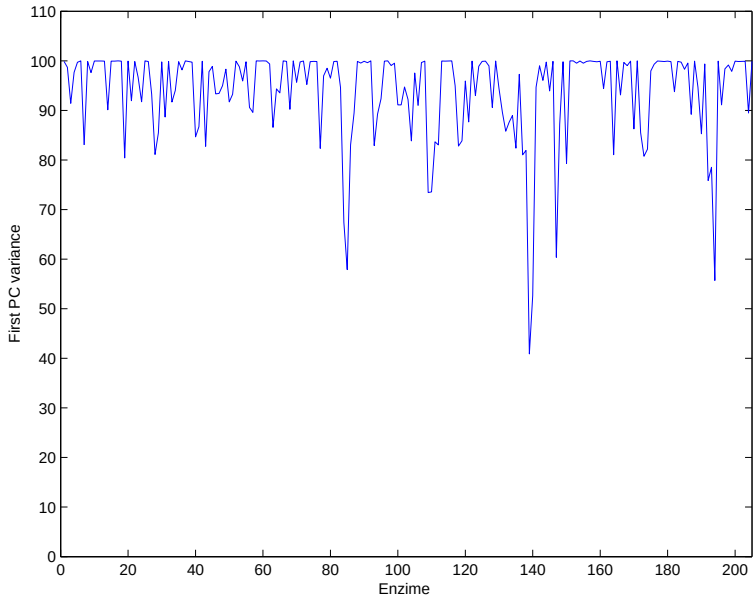

Figura 2.8: Porcentaje de varianza del primer componente principal

porcentaje de reconocimiento falló, el porcentaje de varianza contenido en el primer componente principal era menor al 80 %.

2.5.3. Resultados de la selección de enzimas

Se decidieron utilizar los datos generados por la compresión usando PCA para la creación de la matriz $INPUT$. Se escogió así porque, aunque ambas técnicas de compresión resultaron equivalentes, el PCA arroja algunos datos extra que pueden servir para el análisis posterior de resultados.

El primer paso, aplicar PCA sobre la matriz $INPUT$ de datos comprimidos arrojó los valor de varianza mostrados en la figura 2.9. Pudimos observar que la varianza total de los datos originales esta acumulada en los primeros 47 componentes principales generados. En nuestro caso, desearíamos que de esos 47 componentes principales, alrededor de un 20 % acumularan aproximadamente 80 % de la varianza total, como pasa generalmente en los casos en los que los datos que presentan alta correlación. Esto nos

permitiría tener certeza de que un número reducido de componentes principales será suficiente para clasificar correctamente todos nuestros tipos virales. Sin embargo no fue así, podemos observar que la varianza contenida en cada componente va del 7 % al 0.3 % y el cambio es constante y gradual. La figura 2.10 muestra la varianza acumulada de cada componente. Podemos observar que 80 % de varianza deseado no se alcanza hasta utilizar más de 25 componentes principales.

La razón por la que esto parece un mal indicador es que en muchos casos es necesaria alrededor del 60 % de la varianza de los datos para poder identificar las clases presentes en éstos. Así pues, el hecho de que cada componente contenga una pequeña fracción de la varianza total pudiera indicar que serían necesario un número elevado de k componentes para poder obtener un 100 % de reconocimiento para nuestros 48 tipos virales.

El siguiente paso fue el entrenamiento de las RNA's para la búsqueda de los k componentes principales necesarios para clasificar nuestras muestras. Para cada k se probaron 6 configuraciones de red. La topología se mantuvo como se mencionó en la sección 2.3.2, (k, l, i) donde i se mantiene fijo a 48 salidas. Las topologías varían en el valor de l, que tomó valores de [1-96] neurones, en saltos de 6 unidades. Se escogió este valor por ser divisor de 48, número de tipos virales con los que estamos trabajando, y se sabe como *rule of thumb* que en la capa escondida es recomendable trabajar con una cantidad de neurones similar al número de clases que se desean distinguir .

Pudimos observar que solamente fue necesario $k = 1$ para alcanzar un 100 % de reconocimiento. Es decir, se requirió solamente del primer componente principal para lograr una clasificación completa.

Esto nos indica que en este caso basta una cantidad pequeña de varianza, en relación a la varianza total del sistema, para identificar nuestras muestras. Lo que a su vez habla sobre una gran cantidad de redundancia y información no valiosa en los datos originales.

Continuamos con el proceso seleccionando características con base en su coeficiente

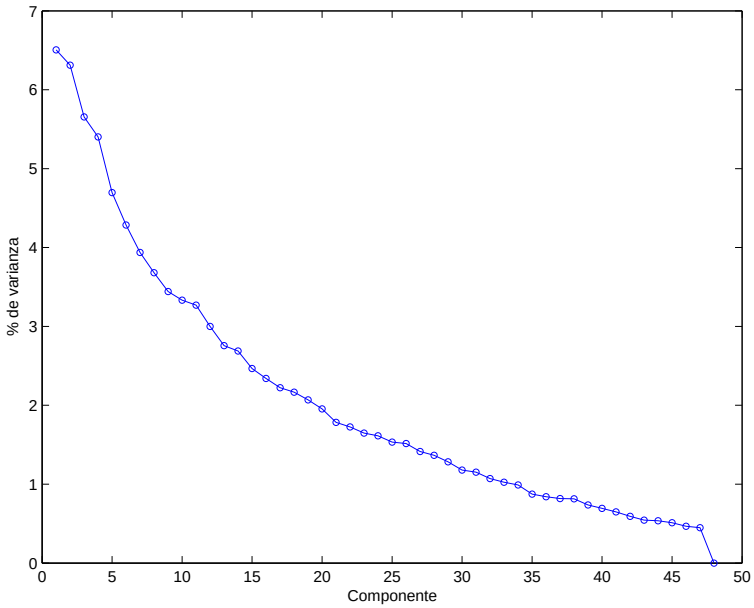

Figura 2.9: Porcentaje de varianza contenida en cada de cada PC al analizar la matriz INPUT

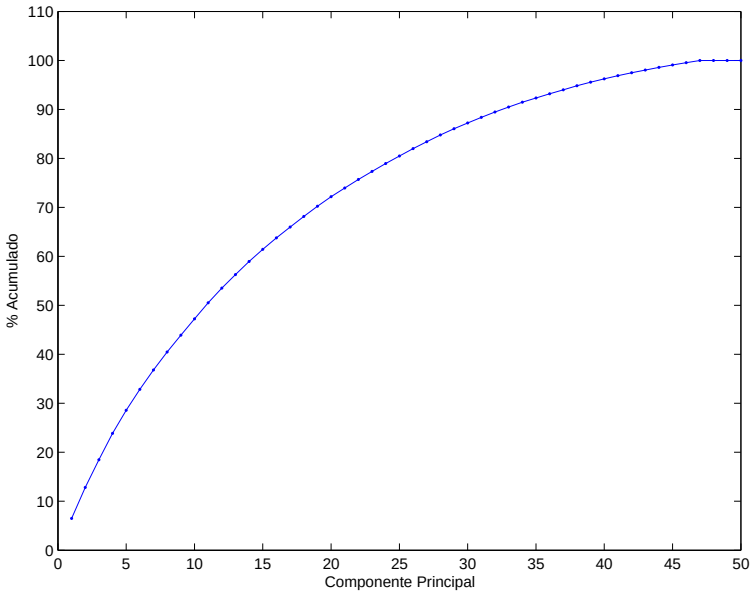

Figura 2.10: Varianza acumulada de los PC's generados al analizar la matriz INPUT

relacionado, como se explicó en la sección 2.3. Observamos que el porcentaje de reconocimiento aumenta con cada adición de una característica a nuestro conjunto de características seleccionadas, sin embargo se logró obtener un conjunto con un 100 % de reconocimiento hasta que el tamaño de éste alcanzó 34 enzimas.

Sabemos que el primer componente principal es suficiente para clasificar nuestros tipos virales, sin embargo observamos que tomar enzimas basándonos solamente en los coeficientes del polinomio para el cálculo de éste no arrojó buenos resultados. Como prueba adicional decidimos probar agregando coeficientes de otros componentes principales, siguiendo el procedimiento descrito.

Utilizamos los polinomios que calculan los componentes principales del 1 al 5. En este caso se logró obtener un 100 % de reconocimiento utilizando 14 enzimas.

Para ver el listado completo de las soluciones vea el apéndice A.

2.6. Conclusiones

2.6.1. Comparación entre compresión con RNA's y PCA

Los resultados obtenidos a través de RNA's cuello de botella y PCA fueron muy similares. En los dos casos los porcentajes de reconocimiento se mantuvieron casi constantes.

Un dato interesante que pudimos observar para casi todas las enzimas fue que las redes cuello de botella convergen a una solución con las mismas propiedades que el primer componente principal. Su solución, aunque numéricamente diferente, mantiene la misma varianza y valores distribuidos de manera similar al primer componente principal.

La figura 2.11 muestra la forma en que se distribuyen los valores de las dos técnicas. La gráfica muestra el valor arrojado por la compresión por PCA y por la compresión por

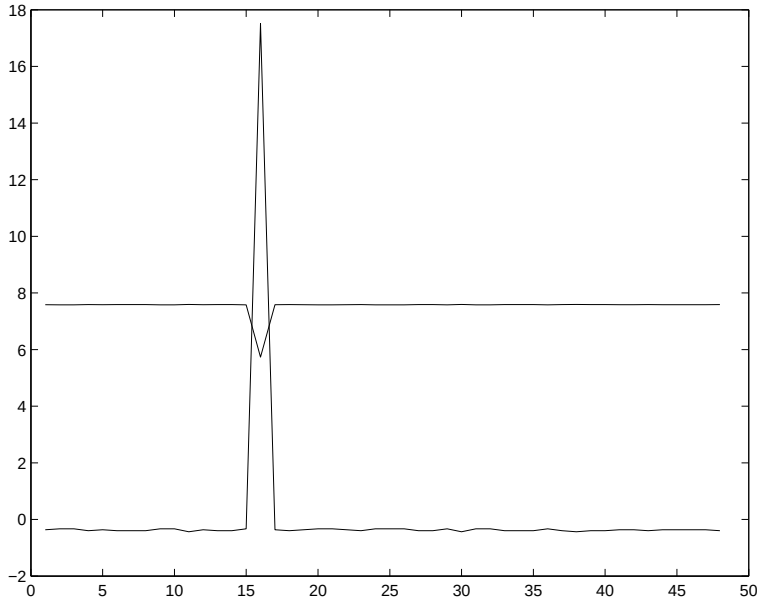

Figura 2.11: Comparación de los valores de salida resultantes de la compresión por RNA y PCA

RNA. Podemos observar que aunque los valores son diferentes, estos se distribuyen de manera similar. Este mismo comportamiento puede observarse para todas las enzimas.

2.6.2. Selección PCA y RNA's

La selección de características utilizando PCA y RNA's tiene la ventaja de ser sencillo de implementar y, si se sustituyen las redes neuronales por un clasificador más simple, resulta de rápida ejecución. Este cambio es necesario pues el tiempo requerido para el entrenamiento de cada red neuronal para la verificación de las selecciones resulta excesivo.

También observamos que el método puede servir de guia para la selección de características, sin embargo los resultados están muy lejos del óptimo.

La razón de esto es que una varianza alta no es suficiente para garantizar optimalidad del conjunto de características seleccionado. Las enzimas seleccionadas fueron

aquellas que más contribuían a maximizar la varianza de los componentes principales seleccionados, pero esto no garantiza que éstas se complementen entre sí.

Los conjuntos formados de esta manera aun mantuvieron un alto grado de redundancia.