CIMAT

Centro de Investigación en Matemáticas, A.C

Un Estudio de la Técnica de Análisis de Componentes Independientes

Tesis

que para obtener el grado de

Maestro en Ciencias con Especialidad en Computación y Matemáticas Industriales

presenta

Ivete Sánchez Bravo

Asesores

Dr. Johan Van Horebeek . CIMAT.

Guanajuato, Gto. Noviembre de 2002.

0.1 AGRADECIMIENTOS

- A Dios por ayudarme durante toda mi vida.
- A mis padres por su apoyo, comprensión y ejemplo de perseverancia para realizar sus metas, pero principalmente por su amor.
- A Ruth y Rodrigo por su presencia y amor en todos los momentos de mi vida.
- A mi asesor, el Dr. Johan Van Horebeek, por todo su esfuerzo y paciencia para la realización de esta tesis.
- A Alonso por compartir conmigo su cariño en ésta bella etapa de nuestras vidas.
- A mis compañeros que me brindaron su amistad y el apoyo necesario en los momentos dificiles de la maestría, en forma especial para David.
- A mis amigos por sus porras y amistad sincera.
- A todo el personal de Cimat, principalmente a los profesores por las horas dedicadas a la educación.
- A Conacyt, ya que sin su apoyo económico no podría haber estudiado este posgrado.

Chapter 1

INTRODUCCIÓN

El análisis de componentes independientes tiene su origen en el problema de recuperar las fuentes originales que conforman a una señal usando únicamente una mezcla lineal no conocida de ellas.

Un ejemplo muy ilustrativo es, teniendo una grabación de un concierto musical, tratar de recuperar los fragmentos interpretados por cada uno de los instrumentos que participaron en la grabación, así como las voces de los cantantes. Otro ejemplo es "el problema de la fiesta de cocktail" (fig. 1.1): Suponemos que en una habitación están dos personas hablando simultáneamente, también en el cuarto se encuentran dos micrófonos ubicados en diferentes posiciones. Cada micrófono se conecta a una grabadora que almacena las voces en cada instante de tiempo t. De ésta manera se tienen dos grabaciones diferentes del mismo cuarto, y en cada una de ellas se pueden escuchar las palabras de ambas personas con diferente intensidad dependiendo la distancia a la que se encontraban del micrófono. El problema ahora es separar las voces de las personas en cada instante de tiempo t utilizando solamente dichas grabaciones. Así, el problema se puede plantear de la siguiente manera:

$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$

$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t)$

donde $x_1(t)$ y $x_2(t)$ son las señales que registran los micrófonos (x_1 y x_2 son las magnitudes y t es el índice del tiempo). Las señales de las voces de las personas se representan por $s_1(t)$ y $s_2(t)$ y finalmente a_{11}, a_{12}, a_{13} y a_{14} son parámetros que dependen de las distancias entre los micrófonos y las personas. En este modelo se omitieron problemas de retraso en las grabaciones

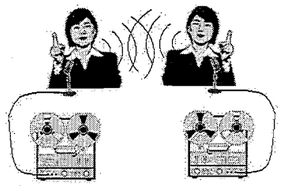


Figure 1-1: Problema de la fiesta de cocktail. Se busca recuperar las voces de las personas que se encuentran en una habitación, únicamente a partir de grabaciones.

y otros factores extras para mayor sencillez.

Actualmente, si nosotros conocemos los parámetros a_{ij}, podemos resolver la ecuación lineal mediante métodos clásicos; pero de no ser así, tenemos un problema mal planteado. Sorprendentemente, asumir como información a priori que las señales $s_1(t)$ y $s_2(t)$, en cada instante t, como estadísticamente independientes, en muchos casos es suficiente para poder resolver el problema.

La técnica de Análisis de Componentes Independientes(ICA) estima los a_{ij} basándose en la suposición de independencia entre $s_1(t)$ y $s_2(t)$. En forma general, si se tienen diferentes señales fuente y se trata de encontrarlas utilizando un conjunto de señales mezcladas, el problema se conoce como Separación de Fuentes a Ciegas (Blind Source Separation, BSS). En la fig.1.2 se muestran el problema de BSS, a partir de las señales mezcladas, se tiene una aproximación de las originales (con un cambio de signo).

Se puede considerar ICA como un modelo basado en los modelos de variables latentes[Apéndice]. Esta técnica es relativamente nueva, fue presentada por primera vez en 1980 en el contexto de redes neuronales artificiales. A mediados de los 90's se introdujeron nuevos algoritmos eficientes por el grupo de la Universidad de Helsinki, junto con demostraciones sorprendentes del efecto en el problema de la fiesta de cocktail, por lo que nuevamente ICA está siendo muy estudiado[1].

El objetivo principal de ésta tesis es presentar un estudio de éste método, ya que por lo nuevo de la técnica no es muy conocida en México; también generar un documento fácil de entender a partir de diferentes bibliografias; resaltando que la investigación de la técnica se empezó desde la parte elemental. Se presentan algunas aplicaciones para clasificación(Capítulo

Part I

PARTE TEORICA

Chapter 2

DEFINICIÓN DEL MODELO DE COMPONENTES INDEPENDIENTES(ICA)

2.1 PLANTEAMIENTO DEL PROBLEMA

El problema de Blind Source Separation consiste en recuperar las señales no observadas o "fuentes", en base a varias mezclas observadas. Por lo general, las observaciones son obtenidas como salidas de un conjunto de sensores, donde cada sensor recibe una diferente combinación de señales fuente. El término "source" significa que se trata de la señal original y "blind" implica que las señales fuentes no son observadas y que no se tiene información disponible de como se hicieron las mezclas. El análisis de componentes independientes (ICA) es muy utilizado para resolver éste tipo de problemas.

Asi, ICA es un método para encontrar factores pesados en el análisis estadístico de datos en múltiples dimensiones. Lo que lo distingue de otros métodos es que se buscan componentes que cumplan con las características de independencia estadística y también de no gausseanidad, condiciones que se explicarán a lo largo de éste capítulo.

A continuación, se definirá el modelo básico: Supongamos que se observan n variables aleatorias de $x_1, \cdots, x_n$ las cuales están modeladas como combinaciones lineales de n variables

aleatorias $s_1, \cdots, s_n$.

$$x_i = a_{i1}s_1 + a_{i2}s_2 + ... + a_{in}s_n \qquad \forall i = 1, \cdots, n \tag{2.1}$$

donde a_{ij} son coeficientes reales, $i, j = 1 \cdots n$. Por definición las s_i son independientes. Los componentes independientes s_j son variables latentes (no pueden ser observadas directamente) y los coeficientes de la mezcla a_{ij} son desconocidos. Utilizando las variables x_i se quieren estimar los a_{ij} y los s_i.

Expresando matricialmente el modelo ICA, tenemos:

$$\mathbf{x} = \mathbf{As} \tag{2.2}$$

donde:

$\mathbf{x}$ es el vector cuyos elementos son las mezclas, x_i.

$\mathbf{A}$ es la matriz que contiene a los coeficientes, a_{ij}

$\mathbf{s}$ es el vector cuyos elementos son las fuentes, s_i

en resumen, se busca una matriz $\mathbf{W}$, la cual, al ser aplicada como una transformación a los datos ($\mathbf{s} = \mathbf{Wx}$), haga que los componentes s_j, $i = 1, \cdots, n$, sean estadísticamente independientes. Si $\mathbf{A}$ fuera invertible, entonces podría expresarse simplemente como $\mathbf{W} = \mathbf{A}^{-1}$.

2.1.1 AMBIGUEDADES

El método de ICA no siempre va a encontrar para $\mathbf{W}$ a la matriz $\mathbf{A}^{-1}$ por las siguientes ambiguedades:

1. No se puede calcular la varianza de cada componente independiente. La razón es que no se conoce ni la $\mathbf{A}$ ni las s_i, por lo que cualquier escalar puede ser cancelado dividiendo la columna correspondiente a_j de $\mathbf{A}$ por el mismo escalar α_j. Como consecuencia de ésto, se asume que la varianza de s_j es unitaria.

$$x_i = \sum_j (\frac{1}{\alpha_j} a_{ij}) s_j \alpha_j \tag{2.3}$$

2. Otra ambiguedad importante, muy relacionada con la anterior, se presenta en el signo, ya que se puede multiplicar un componente por -1 sin alterar el modelo. Esta ambiguedad no afecta en muchas aplicaciones.

3. No se puede determinar el orden de los componentes independientes. Debido a que no se conocen ni s ni $\mathbf{A}$, se puede cambiar el orden de $x = \sum_{i=1}^{n} a_i s_i$ y llamar el "primer" componente al que se quiera.

2.1.2 SUPOSICIONES Y RESTRICCIONES

Existen varias restricciones, que a la vez son consideradas suposiciones, que el modelo ICA tiene que cumplir. En general, la dimensión de los datos observados tiene que ser igual al número de componentes independientes[1,6]. Algunas restricciones importantes son las siguientes:

Ausencia de Ruido

Se asume un modelo completamente libre de ruido. En el caso de ruido aditivo, el modelo de ICA se expresa de la siguiente manera:

$$\mathbf{x} = \mathbf{As} + \mathbf{n} \tag{2.4}$$

donde $\mathbf{n}$ representa el ruido. (Para verificar la estimación de $\mathbf{A}$ con presencia de ruido ir a Apéndice)

Independencia

La suposición básica de ICA, como su nombre lo indica, es que los componentes s_i de los vectores, $\mathbf{s}$, sean independientes unos de los otros. Podría pensarse que la correlación implica independencia, pero no es así. Las variables aleatorias son correlacionadas cuando $E(S_i - E(S_i))(S_j - E(S_j)) = 0$ por lo que, la independencia estadística implica no correlación pero no viceversa. Sólo en el caso de variables gausseanas existe esta doble implicación.

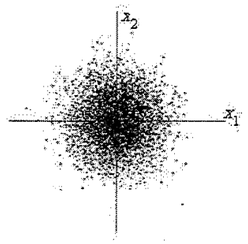


Figure 2-1: Distribución multivariada de dos variables independientes gausseanas.

No gausseanidad

Supongamos que la distribución conjunta de dos componentes independientes, s_1 y s_2 es gausseana:

$$p(s_1, s_2) = \frac{1}{2\pi} \exp(-\frac{s_1^2 + s_2^2}{2}) = \frac{1}{2\pi} \exp(-\frac{||\mathbf{s}||^2}{2}) \tag{2.5}$$

asumiendo que la matriz de mezclas $\mathbf{A}$ es ortogonal, la densidad conjunta de las variables de mezclas está dada por:

$$p(x_1, x_2) = \frac{1}{2\pi} \exp(-\frac{||\mathbf{A}^T\mathbf{x}||}{2})|\det \mathbf{A}^T| \tag{2.6}$$

dada a ortogonalidad de $\mathbf{A}$, se tiene $||\mathbf{A}^T x||^2 = ||x||^2$ y $|\det \mathbf{A}| = 1$ ya que como $\mathbf{A}$ es ortogonal $\mathbf{A} = \mathbf{A}^T$. Entonces se tiene,

$$p(x_1, x_2) = \frac{1}{2\pi} \exp(-\frac{||\mathbf{x}||^2}{2}) \tag{2.7}$$

Como se puede ver, las distribuciones de la matriz de mezclas y de los datos originales son las mismas, por lo que no se puede decir cúal es cual (fig. 2.2). Por eso no se permite que las variables sean gausseanas.

2.2 PREPROCESOS

2.2.1 Centrar las variables

Sin perder generalidad, se puede asumir que tanto las variables de mezcla y los componentes independientes tienen media igual a cero. Si esto no se cumple, se puede realizar un preproceso, restando la media muestral a cada vector, así

$$\mathbf{x}^* = \mathbf{x} - E\{\mathbf{x}\} \tag{2.8}$$

entonces, los componentes independientes tienen media cero,

$$E\{\mathbf{s}\} = \mathbf{A}^{-1}E\{\mathbf{x}\} \tag{2.9}$$

Éste preproceso no afecta a la matriz de mezcla, pero sin embargo es muy útil para simplificar ciertos cálculos en los algoritmos para encontrar los componentes independientes (Capítulo 3). Así que, después de obtener la nueva matriz de mezclas y los componentes independientes para los datos con media cero, se pueden reconstruir añadiendoles $\mathbf{A}^{-1}E\{\mathbf{x}^*\}$ a cada uno.

2.2.2 Blanqueado (Whitening) y Sphering

Dos variables aleatorias y_1 y y_2 se dice que están decorrelacionadas si su covarianza es cero:

$$cov(y_1, y_2) = E\{y_1 y_2\} - E\{y_1\}E\{y_2\} = 0 \tag{2.10}$$

El blanqueado de un vector con media cero significa que sus componentes no están correlacionados y que sus varianzas son iguales a la unidad. En otra palabras, la matriz de covarianza es igual a la matriz identidad.

$$E\{yy^T\} = \mathbf{I} \tag{2.11}$$

en consecuencia, el blanquear una matriz significa multiplicarla por una matriz $\mathbf{V}$ ($\mathbf{z} = \mathbf{V}\mathbf{x}$) tal que el nuevo vector $\mathbf{z}$ sea blanco. También se le conoce como "sphering" y es útil como un preprocesamiento para el método de ICA, la importancia reside en que la nueva matriz de

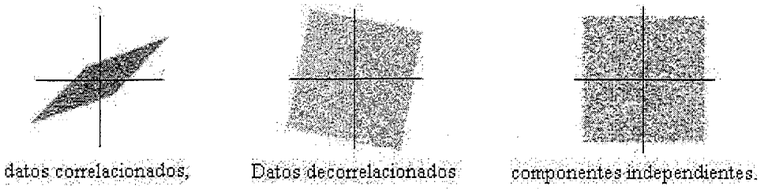


Figure 2-2: Datos después de cada paso importante en el proceso para encontrar los componentes independientes.

mezclas $\tilde{\mathbf{A}} = \mathbf{VA}$ es ortogonal porque:

$$\mathbf{I} = E\{zz^T\} = \tilde{\mathbf{A}}E\{ss^T\}\tilde{\mathbf{A}}^T = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^T \tag{2.12}$$

así se restringe la búsqueda de los componentes a un espacio de matrices ortogonales.Una matriz ortogonal contiene $n(n-1)/2$ grados de libertad, por lo que por ejemplo, en un espacio bidimensional, la búsqueda de una transformación ortogonal se restringe a encontrar un ángulo. En un espacio n-dimensional, una matriz ortogonal contiene solo la mitad del número de parámetros de una matriz arbitraria [ref.1]. La (fig. 2.2) muestra los datos centrados originales (correlacionados), después de aplicarles el blanqueado y con ICA.

Una manera de realizar la transformación de blanqueado es mediante la descomposición por eigenvalores (SVD) de la matriz de covarianza.

$$E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{E}\mathbf{D}\mathbf{E}^T \tag{2.13}$$

donde $\mathbf{E}$ es la matriz ortogonal de eigenvectores de $\mathbf{E}\{\mathbf{x}\mathbf{x}^T\}$ y $\mathbf{D}$ es la matriz diagonal de sus eigenvalores, $\mathbf{D} = diag(d_1, ..., d_n)$. La matriz de blanqueado es entonces:

$$\mathbf{V} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T \tag{2.14}$$

donde la matriz $\mathbf{D}^{-1/2} = diag(d_1^{-1/2}, ..., d_n^{-1/2})$.

Ahora, supongamos que el modelo ICA está blanqueado. Se tendría

$$\mathbf{z} = \mathbf{VAs} = \tilde{\mathbf{A}}\mathbf{s} \tag{2.15}$$

Resolver el problema "blanqueado" es similar a resolver el problema de independencia, ya que estos dos conceptos están relacionados, pero NO es suficiente para lo que deseamos. Supongamos una transformación ortogonal $\mathbf{U}$ de $\mathbf{z}$:

$$\mathbf{y} = \mathbf{U}\mathbf{z} \tag{2.16}$$

Dada la ortogonalidad de $\mathbf{U}$, se tiene

$$E\{\mathbf{y}\mathbf{y}^T\} = E\{\mathbf{U}\mathbf{y}\mathbf{y}^T\mathbf{U}^T\} = \mathbf{U}\mathbf{I}\mathbf{U}^T = \mathbf{I} \tag{2.17}$$

en otras palabras, $\mathbf{y}$ es blanco también, por lo que no se puede decir si los componentes independientes están dados por $\mathbf{z}$ o por $\mathbf{y}$ si unicamente utilizamos la propiedad de blancura.

Chapter 3

MÉTODOS PARA CALCULAR LOS COMPONENTES INDEPENDIENTES

3.1 INTRODUCCIÓN

Una vez definido el modelo, se tiene que encontrar a la matriz $\mathbf{W}$ que transforme los datos observados (ya preprocesados) en los componentes independientes $\mathbf{s}$(ya que $\mathbf{s} = \mathbf{Wz}$). Para encontrar $\mathbf{W}$, el método usual es formular una función objetivo $O(\mathbf{w})$ que mide la independencia entre los componentes y después utilizar algún algoritmo para optimizarla (Fig. 3.1). Así,

método de ICA = Función Objetivo + Algoritmo de Optimización

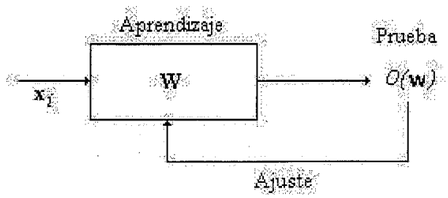


Figure 3-1: Diagrama a bloques del proceso de optimización.

El algoritmo de optimización y la función objetivo determinan las propiedades del método de ICA de la siguiente manera:

1. las propiedades estadísticas: consistencia y robustez, dependen de la elección de la función objetivo[7].

2. las propiedades algorítmicas como velocidad de convergencia, requerimientos de memoria y estabilidad numérica, dependen del bloque de optimización [1].

Además, se tiene que escoger si se desean estimar los componentes independientes uno por uno (calculando en cada paso un renglón de $\mathbf{W} = [w_1, w_2, \ldots, w_n]$) o en paralelo (toda la matriz $\mathbf{W}$). La primera manera, descrita en la sección 3.2, abarca los algoritmos conocidos como de una unidad, que comprenden las técnicas de medición de no gausseanidad por medio de kurtosis y negentropía. Se les conoce como algoritmos de una unidad porque solo calculan un componente independiente por cada ejecución, por lo que el calcular varios componentes independientes, es correr varias veces el algoritmo hasta encontrar el número de componentes que se desean.

Finalmente, en el última sección del capítulo se presentan los que calculan toda la matriz al mismo tiempo. En todos los métodos para construir $O(\mathbf{w}^T\mathbf{z})$ se utilizan los algoritmos de gradiente y punto fijo.

3.2 ICA BASADO EN LA MEDICIÓN DE LA NO GAUSSEANIDAD

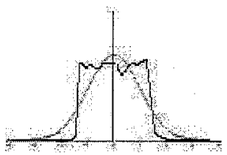
La no gausseanidad tiene gran importancia para la estimación del modelo ICA. Basándonos en el Teorema del Límite Central puede decirse, que la suma de dos o mas variables aleatorias que no son normales tiene una distribución mas cercana a la normal que cualquiera de las variables aleatorias originales.

Suponiendo que se tiene un vector de datos $\mathbf{x}$ que se distribuye de acuerdo al modelo ICA:

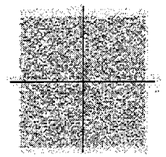
$$\mathbf{x} = \mathbf{As} \tag{3.1}$$

entonces, el estimar uno de los componentes independientes es calcular una combinación lineal de los x_i.

$$\mathbf{y} = \mathbf{b}^T\mathbf{x} = \mathbf{b}^T\mathbf{As} = \mathbf{q}^T\mathbf{s}$$



Densidad empírica de un componente independiente, comparada con una distribución gausseana



conjunta de dos componentes independientes

Figure 3-2: Distribución conjunta y densidad de dos componentes independientes con distribución uniforme.

donde $\mathbf{b}^T\mathbf{x}$ es uno de los componentes independientes.

La idea general es variar los coeficientes de $\mathbf{q}$, ya que se sabe que $\mathbf{y} = \mathbf{q}^T\mathbf{s}$ es usualmente más gausseana que cualquiera de las s_i y menos cuando es igual a una de ellas (siempre y cuando $\mathbf{s}$ sea no gausseana). En práctica, no se conocen los valores de $\mathbf{q}$, pero por definición $\mathbf{q}^T\mathbf{s} = \mathbf{b}^T\mathbf{x}$, por lo que se puede variar $\mathbf{b}$, que es un vector que maximiza la no gausseanidad. Ahora, existe un $\mathbf{b}$ tal que $\mathbf{A}^T\mathbf{b} = q_i$, por lo que $\mathbf{y} = \mathbf{b}^T\mathbf{x} = \mathbf{q}^T\mathbf{s}$ es igual a un componente independiente. Por lo tanto, maximizando la no gausseanidad de $\mathbf{b}^T\mathbf{x}$, se obtienen los componentes independientes.

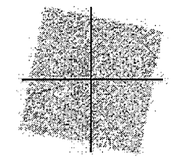
Por ejemplo, considerando dos observaciones que tienen distribuciones uniformes con media cero, como se muestra en la Fig. 3.2. Si éstas variables son mezcladas linealmente y las mezclas son preprocesadas con blanqueado, entonces

$$\mathbf{z} = \mathbf{V}\mathbf{x} = \mathbf{V}\mathbf{A}\mathbf{s} \tag{3.2}$$

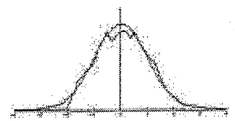
La distribución y la densidad de las variables se muestra en la figura 3.3.

Como se puede ver ésta última densidad es mucho más cercana a la gausseana que la anterior. Encontrando la rotación que nos lleva a los originales componentes independientes, se maximiza la no gausseanidad.

A continuación, se presentan algunas medidas de no gausseanidad.



Densidad conjunta de dos mezclas blanqueadas de componentes independientes



Densidad marginal de la mezcla gausseana

Figure 3-3: Densidad marginal y distribución de dos mezclas blanqueadas de componentes independientes uniformes.

3.2.1 MEDIDA DE LA GAUSSEANIDAD POR KURTOSIS

La kurtosis de una variable aleatoria y que tiene media igual a cero, denotada por $kurt(y)$, está definida por

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 \tag{3.3}$$

Algunas de sus propiedades importantes cuando se tienen dos variables aleatorias independientes son:

$$kurt(x_1 + x_2) = kurt(x_1) + kurt(x_2) \tag{3.4}$$

$$kurt(\alpha x_1) = \alpha^4 kurt(x_1) \tag{3.5}$$

donde α es una constante.

Asumiendo que y está normalizada (varianza unitaria) y partiendo de (3.3), se puede simplificar a:

$$kurt(y) = E\{y^4\} - 3 \tag{3.6}$$

que es una versión del cuarto momento $E\{y^4\}$.

La kurtosis de una variable gausseana es igual a 0; para la mayoria de las variables aleatorias

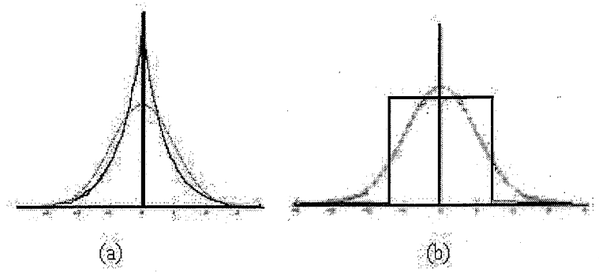


Figure 3-4: Densidad de la distribución de (a) Laplaciano, (b) uniforme. Con lineas punteadas se encuentra la gauseana para comparación.

no gauseanas, es diferente de cero; por lo que es una medida de la no gausseanidad. La kurtosis puede ser positiva o negativa, por lo que la no gausseanidad es medida por su absoluto o por su valor al cuadrado. Las variables aleatorias que tienen kurtosis negativa son llamadas subgausseanas (platykurtic) y las que tienen kurtosis positiva son conocidas como supergausseanas (leptokurtic). Las variables supergauseanas, tienen comunmente funciones de distribución de probabilidad muy puntiagudas, con colas pesadas.Un ejemplo típico es la distribución Laplaciana (Fig. 3.4a), cuya función de densidad de probabilidad está dada por:

$$p(y) = \frac{1}{\sqrt{2}} \exp(-\sqrt{2}|y|) \tag{3.7}$$

Las variables subgausseanas son en general más planas que la normal; un ejemplo es la uniforme (Fig 3.4b):

$$p(y) = \left\{ \begin{array}{l} \frac{1}{2\sqrt{3}}, \text{si } |y| \leq \sqrt{3}; \\ 0 \text{ de otra manera} \end{array} \right\}$$

Para ilustrar como son encontrados los componentes independientes por la maximización de la kurtosis, supongamos un modelo de 2D, $\mathbf{x} = \mathbf{As}$. Asumiendo que los componentes independientes s_1 y s_2 tienen valores de kurtosis $kurt(s_1)$ y $kurt(s_2)$ diferentes de cero y varianzas unitarias. Se busca uno de los componentes independientes $y = \mathbf{b}^T\mathbf{x}$. Sabemos que $y = \mathbf{b}^T\mathbf{x} = \mathbf{b}^T\mathbf{As} = \mathbf{q}^T\mathbf{s} = q_1 s_1 + q_2 s_2$. y por lo tanto

$$kurt(y) = kurt(q_1 s_1) + kurt(q_2 s_2) = q_1^4 kurt(s_1) + q_2^4 kurt(s_2)$$

partiendo de la suposición que $var(y) = 1$, ésto implica que $E\{y^2\} = q_1^2 + q_2^2 = 1$. Geométricamente, esto significa que el vector $\mathbf{q}$ esta restringido al círculo unitario en el plano 2D. Entonces, el problema de optimización se reduce a $max\,|kurt(y)| = \left|q_1^2 kurt(s_1) + q_2^2 kurt(s_2)\right|$, en el círculo unitario. Si las kurtosis son iguales a 1, entonces se tiene: $F(\mathbf{q}) = q_1^4 + q_2^4$ sujeto a la restricción de que $q_1^2 + q_2^2 = 1$. Obteniendose los puntos iguales a (1,0), (-1,0), (0,1) y (0,-1) que son $\pm s_i$.

ALGORITMO DE GRADIENTE

Prácticamente, para maximizar el valor absoluto de la kurtosis, se inicia con un vector $\mathbf{w}$, calculado en la dirección del valor absoluto de la kurtosis de $y = \mathbf{w}^T\mathbf{z}$, basándose en las muestras $\mathbf{z}(1), \ldots, \mathbf{z}(n)$ del vector $\mathbf{z}$ y posteriormente se mueve el vector $\mathbf{w}$ en esa dirección.

El gradiente del valor absoluto de la kurtosis $\mathbf{w}^T\mathbf{z}$ se calcula como:

$$\frac{\delta|kurt(\mathbf{w}^T\mathbf{z})|}{\delta\mathbf{w}} = 4sign\left(kurt\left(\mathbf{w}^T\mathbf{z}\right)\right)\left[E\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\} - 3\mathbf{w}||\mathbf{w}||^2\right] \tag{3.8}$$

(demostración en el Apéndice A). Se busca optimizar sobre una esfera unitaria, por lo que en cada paso del algoritmo el vector tiene que ser dividido entre su norma. El término $3\mathbf{w}||\mathbf{w}||^2$ implica una modificación a la magnitud del vector y no a su dirección; de manera similar que el 4 que multiplica al signo de la kurtosis, por lo que ambos elementos se pueden excluir de la ecuación anterior obteniéndose:

$$\Delta\mathbf{w} \propto sign\left(kurt\left(\mathbf{w}^T\mathbf{z}\right)\right) E\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\} \tag{3.9}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{3.10}$$

Una nueva versión del algoritmo se puede obtener omitiendo el operador de esperanza (gradiente estocástico) y tomando cada observación $\mathbf{z}(i)$ una sola vez. Resultando así:

$$\Delta \mathbf{w} \propto sign\left(kurt\left(\mathbf{w}^T\mathbf{z}\right)\right)\mathbf{z}(\mathbf{w}^T\mathbf{z})^3 \tag{3.11}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{3.12}$$

El algoritmo que se obtiene es el siguiente:

1. Centrar los datos para que su media sea cero
2. Blanquear los datos para obtener $\mathbf{z}$
3. Para cada una de las observaciones $\mathbf{z}(n)$
   (a) Calcular $\Delta \mathbf{w} = sign\left(kurt\left(\mathbf{w}^T\mathbf{z}(n)\right)\right)\mathbf{z}(n)(\mathbf{w}^T\mathbf{z}(n))^3$
   (b) Hacer $\mathbf{w} \leftarrow \mathbf{w} + paso * \Delta \mathbf{w}$
   (c) Normalizar $\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||$
4. Si no converge, regresar al paso 3.

donde el tamaño de paso es definido por el usuario, asegurando que en cada iteración $kurt\left(\mathbf{w}_{t+1}^T\mathbf{z}\right) > kurt\left(\mathbf{w}_t^T\mathbf{z}\right)$. La ventaja de este algoritmo es que tiene una adaptación rápida en un ambiente dinámico. Las desventajas principales son que la convergencia es lenta y depende mucho del punto inicial.

ALGORITMO DE PUNTO FIJO

El algoritmo de punto fijo es una alternativa para mejorar la velocidad de convergencia a la solución esperada. Está fundamentado en la existencia de un punto estable en el método del gradiente.

Demostración: Asumiendo que se desea maximizar la función $F(\mathbf{w}^T\mathbf{z})$ en la esfera unitaria, es decir, bajo la restricción $G(\mathbf{w}) = ||\mathbf{w}|| = 1$. El gradiente de G es igual a $||\mathbf{w}||^2 - 1$, resolviendo por el método de Lagrange se tiene que $\nabla \mathbf{F} = \lambda \nabla \mathbf{G}$; que es:

$$L(w, \lambda) = -F(\mathbf{w}^T\mathbf{z}) + \lambda\left(||\mathbf{w}||^2 - 1\right)$$

por lo que en el máximo, el gradiente de $\mathbf{F}$ apunta en la misma dirección que $\mathbf{w}$. En otras palabra el gradiente es igual a $\mathbf{w}$ multiplicado por una constante escalar.

El gradiente apunta en la dirección de $\mathbf{w}$, lo cual significa que es igual a $\mathbf{w}$ multiplicada por alguna constante escalar. Partiéndose de la ecuación (3.8):

$$\Delta \mathbf{w} \propto \left[E\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\} - 3\mathbf{w}||\mathbf{w}||^2\right] \tag{3.13}$$

donde $||\mathbf{w}||^2 = 1$, por lo que tiene un algoritmo de punto fijo de la siguiente manera:

$$\mathbf{w} \leftarrow E\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\} - 3\mathbf{w} \tag{3.14}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{3.15}$$

El vector final $\mathbf{w}$ da uno de los componentes independientes como una combinación lineal de $\mathbf{w}^T\mathbf{z}$. La convergencia de éste método se determina cuando el valor de $\mathbf{w}$ anterior es igual al $\mathbf{w}$ nuevo; esto es, el producto punto entre ambos vectores es 1. Esto no necesariamente dice que el vector converge al mismo punto, ya que $\mathbf{w}$ y -$\mathbf{w}$ definen la misma dirección. Se ha comprobado que éste metodo trabaja muy bien y que la convergencia es bastante rápida y confiable [ref].

3.2.2 MEDIDA DE LA GAUSSEANIDAD POR NEGENTROPIA

En la sección anterior se utilizó a la kurtosis como una medida de la no gausseanidad. Una de sus desventajas es que es muy sensible a outliers, por lo que no es un medidor robusto.

La entropía es, como la varianza, una medida de la variabilidad. Está relacionada con la información que proporciona una variable aleatoria; aquellas que dan mas información son las que tienen entropía mas grande. La entropía (diferencial) H de un vector aleatorio $\mathbf{y}$ con densidad $p_y(\boldsymbol{\eta})$ está definida como:

$$H(\mathbf{y}) = -\int p_y(\boldsymbol{\eta}) \log p_y(\boldsymbol{\eta}) d\boldsymbol{\eta} \tag{3.16}$$

Una variable gausseana tiene la mayor entropía del conjunto de densidades con una varianza dada, mientras que la distribución uniforme es la distribución que tiene maxima entropía sobre un intervalo acotado. Esto puede ser tomado como que la entropía es una medida de la no

gausseanidad. La entropía es pequeña para variables que están concentradas en unos pocos valores (pdf's muy "puntiagudas").

Para obtener una medida de la no gausseanidad que sea cero para variables gausseanas y siempre no negativa, se utiliza una versión normalizada de la entropía diferencial conocida como la negentropía, la cual se define como:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{3.17}$$

donde $\mathbf{y}_{gauss}$ es una variable gausseana con la misma varianza que $\mathbf{y}$. La ventaja de ésta medida es que se puede demostrar que es el estimador óptimo de la no gausseanidad en términos de la robustez[1]; el problema es que calcularla es sumamente dificil, por lo que se aproxima por medio de polinomios.

A continuación, si $\mathbf{y}$ tiene media cero y varianza unitaria, se va a suponer que $p_y(\xi)$ es casi una densidad gausseana estándar, $\varphi(\xi) = \exp(-\xi^2/2)/\sqrt{2\pi}$. Definiendo los polinomios ortogonales de Chevyshev-Hermite (denotados por H_i donde i es un entero no negativo e indica el orden), por las derivadas de $\varphi(\xi)$, se tiene:

$$\frac{\delta^i \varphi(\xi)}{\delta \xi^i} = (-1)^i H_i(\xi)\varphi(\xi)$$

Realizando una expansión con ellos, se representa un sistema ortonormal en el sentido de que:

$$\int \varphi(\xi) H_i(\xi) H_j(\xi) d\xi = \left\{ \begin{array}{l} 1 \text{ si } i = j \\ 0 \text{ si } i \neq j \end{array} \right\}$$

así, la expansión de Gram-Charlier para $p_y(\xi)$ está dada por:

$$p_y(\xi) \approx \hat{p}_y(\xi) = \varphi(\xi)\left(1 + \kappa_3(y)\frac{H_3(\xi)}{3!} + \kappa_4(y)\frac{H_4(\xi)}{4!}\right)$$

donde $\kappa_3(y) = E\{y^3\}$, $\kappa_4(y) = E\{y^4\} - 3$. Sustituyendo lo anterior en la ecuación de entropia y realizando la aproximación de que $\log(1+\epsilon) \approx \epsilon - \epsilon^2/2$:

$$\begin{aligned} H(y) \approx & -\int \hat{p}_y(\xi)\log\hat{p}_y(\xi)d\xi = \\ & -\int \varphi(\xi)\left(1 + \kappa_3(y)\frac{H_3(\xi)}{3!} + \kappa_4(y)\frac{H_4(\xi)}{4!}\right) \\ & \left[\log\varphi(\xi) + \kappa_3(y)\frac{H_3(\xi)}{3!} + \kappa_4(y)\frac{H_4(\xi)}{4!} - \left(\kappa_3(y)\frac{H_3(\xi)}{3!} + \kappa_4(y)\frac{H_4(\xi)}{4!}\right)^2/2\right] d\xi \end{aligned}$$

simplificando:

$$H(y) \approx -\int \varphi(\xi)\log\varphi(\xi)\,d\xi - \frac{\kappa_3(y)^2}{2\text{x}3!} - \frac{\kappa_4(y)^2}{2\text{x}4!}$$

Finalmente, la aproximación de la negentropía de una variable aleatoria estándar es:

$$J(y) \approx \frac{1}{12}E\{y^3\}^2 + \frac{1}{48}kurt(y)^2 \tag{3.18}$$

Se pueden reemplazar las funciones polinomiales y^3 y y^4por otras funciones G^i(donde i es un índice, no una potencia), por lo que el método proporciona una manera de aproximar la negentropía por medio de las esperazas $E\{G^i(y)\}$ (Demostración en el Apéndice A.1). Si se toman dos funciones no cuadráticas G^i y G^i tal que la primera función es impar y la segunda par, se tiene la siguiente aproximación [1]:

$$J(y) \approx k_1\left(E\left\{G^1(y)\right\}\right)^2 + k_2\left(E\left\{G^2(y)\right\} - E\left\{G^2(\nu)\right\}\right)^2 \tag{3.19}$$

donde k_1 y k_2 son constantes positivas y ν es una variable gausseana con media cero y varianza unitaria. Se ha demostrado que se pueden escoger funciones G que mejoren la aproximación que se dió anteriormente[1]. En especial, las siguientes funciones son bastante utilizadas

$$G_1(y) = \frac{1}{a_1}\log\cosh a_1 y \tag{3.20}$$

$$G_2(y) = -\exp(-y^2/2) \tag{3.21}$$

donde $1 \leq a_1 \leq 2$ (Fig. 3.5).

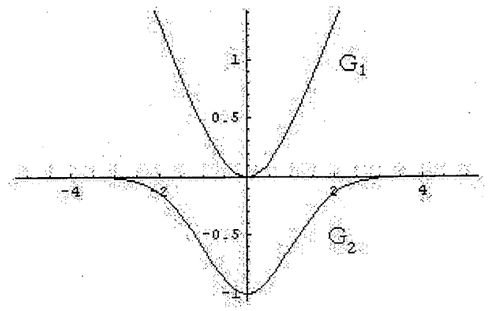


Figure 3-5: Funciones G's utilizadas para aproximar la negentropia.

En el caso que se utilice una sola función no cuadrática G, la aproximación que se tiene es:

$$J(y) \propto [E\{G(y)\} - E\{G(\nu)\}]^2 \tag{3.22}$$

ALGORITMO DE GRADIENTE

Al igual que en el caso de la kurtosis, lo que se busca es maximizar la negentropia. Realizando la maximización del algoritmo al igual que para la kurtosis(ver Apéndice A). Se tiene:

$$\Delta \mathbf{w} \propto \gamma E\{\mathbf{z} g(\mathbf{w}^T \mathbf{z})\} \tag{3.23}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{3.24}$$

donde $\gamma = E\{G(\mathbf{w}^T\mathbf{z})\} - E\{G(\nu)\}$, ν es una variable aleatoria gauseana estándar y g es la derivada de la funcion G. La esperanza será omitida para obtener un algoritmo de gradiente en linea.

$$\Delta\gamma = E\{G(\mathbf{w}^T\mathbf{z})\} - E\{G(\nu)\} - \gamma \tag{3.25}$$

donde γ es una medida de adaptación "automática" que es similar al signo de la kurtosis en el algoritmo basado en ésta medida.

El algoritmo en forma general es el siguiente:

1. Centrar los datos para que su media sea igual a cero
2. Blanquear los datos para obtener $\mathbf{z}$
3. Escoger un vector inicial $\mathbf{w}$(puede ser aleatoriamente), y un valor inicial para γ
4. Calcular $\Delta\mathbf{w} \propto \gamma E\{\mathbf{z} g(\mathbf{w}^T\mathbf{z})\}$
5. Normalizar $\mathbf{w}$
6. Si el signo de γ no es parte del conocimiento a priori, calcularlo como $\Delta\gamma = (G(\mathbf{w}^T\mathbf{z}) - E\{G(\nu)\}) - \gamma$
7. Si no converge, regresar al paso 4.

La función g es la derivada de la función G correspondiente, de esta manera:

$$g_1(y) = \tanh(a_1 y) \tag{3.26}$$

$$g_2(y) = y \exp(-y^2/2) \tag{3.27}$$

$$g_3(y) = y^3 \tag{3.28}$$

donde $1 \leq a_1 \leq 2$ (Fig. 3-6).

ALGORITMO DE PUNTO FIJO (FASTICA)

El algoritmo de punto fijo, al igual que el introducido para el caso de la kurtosis, es mucho más rápido y estable que el de gradiente (deducción Apéndice A) y es el siguiente:

1. Centrar los datos para hacer que su media sea cero.

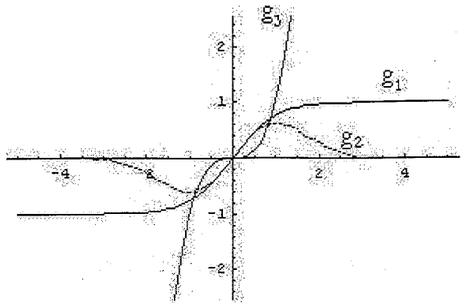


Figure 3-6: Derivadas de las funciones G's

2. Blanquear los datos para obtener $\mathbf{z}$

3. Escoger un vector inicial $\mathbf{w}$(puede ser aleatoriamente).

4. Sea $\mathbf{w} \leftarrow E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z}) - E\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w}\right\}$ donde g está definida como el en algoritmo anterior y g' de la siguiente manera: $g_1'(y) = a_1(1 - \tanh^2(a_1 y)$; $g_2'(y) = (1 - y^2)\exp(-y^2/2)$; $g_3'(y) = 3y^2$

5. Normalizar $\mathbf{w}$

6. Si no converge, regresar al paso 4.

Como criterio de convergencia, se verifica que el nuevo y el viejo valor de $\mathbf{w}$ apunten en la misma dirección; esto es, que el producto punto sea igual a 1.No es necesario que convergan al mismo punto , ya que $\mathbf{w}$ y - $\mathbf{w}$ apuntan a la misma dirección.

PROPIEDADES DEL ALGORITMO DE FASTICA

El algoritmo de FastICA tiene algunas propiedades importantes comparándolo con otros métodos[5].

1. La convergencia es cúbica (o al menos cuadrática). Ésta es una diferencia con los algoritmos basados en métodos de descenso de gradiente, donde la convergencia es únicamente

lineal. Ésto significa que la convergencia es muy rápida, como se ha confirmado en varios experimentos[5].

2. Al contrario que los algoritmos basados en gradiente, no hay parámetros que ajustar, por lo que el algoritmo es fácil de utilizar.

3. El desempeño del método puede ser optimizado al escoger la nolinealidad g. En particular, se pueden tener algoritmos que son robustos y/o de minima variabilidad.

4. Los componentes independientes pueden ser calculados uno por uno, lo cual es fuertemente equivalente a realizar projection pursuit (ver Capítulo 3). Ésto es útil para análisis exploratorio de datos, reduce la carga computacional cuando solo se necesita calcular algunos componentes independientes.

5. El método de FastICA tiene las muchas ventajas se desarrolla en paralelo, está distribuido, es simple computacionalmente y requiere poco espacio en memoria. Los métodos de gradiente estocástico, son preferibles solo cuando se necesita un rápida adaptabilidad en un ambiente.

3.2.3 ESTIMACIÓN DE VARIOS COMPONENTES INDEPENDIENTES

Para calcular varios componentes independientes, se necesita correr un algoritmo de una unidad varias veces hasta encontrar el numero de componentes que se desean. Para extender dichos métodos a algoritmos que calculen varios componentes, se parte del principio de que los componentes independientes $\mathbf{w}_i$ tienen que estar no correlacionados. En un espacio blanqueado se tiene: $E\left\{\left(\mathbf{w}_i^T\mathbf{z}\right)\left(\mathbf{w}_j^T\mathbf{z}\right)\right\} = \mathbf{w}_i^T\mathbf{w}_j$, y como consecuencia ortogonalidad. Los $\mathbf{w}_i$ son los renglones de la inversa de la matriz de mezclas, y son iguales a las columnas de la matriz de mezclas por la propiedad $\mathbf{A}^{-1} = \mathbf{A}^T$. Se puede utilizar alguno de los siguientes algoritmos.

ORTOGONALIZACIÓN DESINFLANTE (Deflationary orthogonalization)

Una manera de ortogonalizar es utilizando el método desinflante con GramSchmidt; ésto significa que se van calculando uno por uno cada componente. Cuando ya se calcularon p componentes independientes (vectores $\mathbf{w}_1, ..., \mathbf{w}_p$), se ejecuta un algoritmo de una unidad para

encontrar el componente $\mathbf{w}_{p+1}$, después de cada iteración se resta de $\mathbf{w}_{p+1}$ la proyección $\left(\mathbf{w}_p^T \mathbf{w}_j\right) \mathbf{w}_j, j = 1, ..., p$ de los p vectores calculador anteriormente y se renormaliza $\mathbf{w}_{p+1}$. El algoritmo es el siguiente:

1. Escoger m, el número de componentes que se desean estimar. Hacer $p \leftarrow 1$
2. Inicializar $\mathbf{w}_p$. (puede ser aleatoriamente)
3. Hacer una iteración de un algoritmo de una unidad para obtener $\mathbf{w}_p$
4. Hacer la siguiente ortogonalización:

$$\mathbf{w}_p \leftarrow \mathbf{w}_p - \sum_{j=1}^{p-1} \left(\mathbf{w}_p^T \mathbf{w}_j\right) \mathbf{w}_j \tag{3.29}$$

   Normalizar $\mathbf{w}_p$ dividiendolo por su norma

5. Si $\mathbf{w}_p$ no converge, regresar al paso 3.
6. Hacer $p \leftarrow p+1$.Si p no es más grande que el número de componentes deseados, regresar al paso 2.

ORTOGONALIZACIÓN SIMÉTRICA

Este tipo de ortogonalización se utiliza cuando no se desea que alguno de los vectores sea "privilegiado" sobre los demás. Ésto es, todos los vectores son estimados en paralelo en lugar de uno por uno. Una ventaja sobre el anterior es que el algorimo de ortogonalización desinflante acarrea el error del primer componente sobre el segundo y asi sucesivamente. Otra ventaja es que se emplean calculos en paralelo.

A grandes razgos, el método consiste en hacer pasos iterativos de algoritmos de una unidad para cada vector $\mathbf{w}_i$ en paralelo y después ortogonalizar todas las $\mathbf{w}_i$ por un método simétrico, de la siguiente manera:

1. Escoger el número de componentes independientes m a estimar.
2. Inicializar los $\mathbf{w}_i$, $i = 1, ..., m$ (puede ser aleatoriamente)

3. Hacer una iteración de un algoritmo de un solo paso para cada $\mathbf{w}_i$ en paralelo.
4. Realizar una ortogonalización simétrica de la matriz $\mathbf{W} = (\mathbf{w}_1, ...\mathbf{w}_m)^T$.
5. Si no converge, regresar al paso 3.

La ortogonalización simétrica puede hacerse de las siguientes maneras:

PRIMERA FORMA

1. Hacer $\mathbf{W} \leftarrow \mathbf{W}/||\mathbf{W}||$.
2. Hacer $\mathbf{W} \leftarrow \frac{3}{2}\mathbf{W} - \frac{1}{2}\mathbf{W}\mathbf{W}^T\mathbf{W}$,
3. Si $\mathbf{W}\mathbf{W}^T$ no ésta lo suficientemente cerca de la identidad, ir al paso 2.

SEGUNDA FORMA:

Hacer $\mathbf{W} \leftarrow \left(\mathbf{W}\mathbf{W}^T\right)^{-1/2}\mathbf{W}$; donde $\left(\mathbf{W}\mathbf{W}^T\right)^{-1/2}$ se obtiene mediante la descomposición en eigenvalores de

$$\mathbf{W}\mathbf{W}^T = \mathbf{E} diag\left(d_1^{-1/2}, ..., d_m^{-1/2}\right) \mathbf{E}^T. \tag{3.30}$$

Los métodos anteriores se obtuvieron de [1].

3.2.4 CRITERIOS DE DESEMPENO

Para evaluar el desempeño de un algoritmo, se utilizan datos generados por alguna fuente conocida y así cotejar el error del resultado estimado con el verdadero. Una manera de comparar algorimos es mediante el desempeño estadístico con una matriz de permutación[1] de la siguiente manera:

$$E_1 = \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1 \right) + \sum_{j=1}^{N} \left(\sum_{i=1}^{N} \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1 \right) \tag{3.31}$$

donde p_{ij} es el ij-ésimo elemento de la matriz de permutacion $\mathbf{P} = \mathbf{BA}$. Si los componentes independientes están separados perfectamente, $\mathbf{P}$ es una matriz de permutación (donde los elementos tienen signos diferentes). Una matriz de permutación está definida como aquella en la que en cada renglón y en cada columna, solo un elemento es diferente a cero. El criterio

de desempeño E_1 da un valor casi cero mientras más cercano esté **P** de ser una matriz de permutación.Otro criterio, conocido como E_2 es similar al anterior, con la diferencia de que se toman los valores p_{ij}^2.

3.2.5 RESULTADOS PRÁCTICOS

En esta sección, se realizaron pruebas con los métodos basados en la no gausseanidad, tanto de gradiente como de punto fijo. La ortogonalidad de los componentes se hizo con el método desinflante. Los datos que se utilizaron fueron en 2D y se generaron a partir de diferentes distribuciones.

Comparación del tiempo de ejecución entre los métodos

El objetivo de ésta sección fue revisar el tiempo de ejecución en promedio que tarda cada uno de los métodos utilizando diferentes tipos de distribuciones(lo mas lejanas posibles a una gausseana) para generar los datos de entrada.

Características de la prueba

(a) Puntos de arranque aleatorios

(b) 1000 muestras. En cada corrida se utiliza el mismo conjunto de datos

(c) 40 corridas por método

(d) Métodos programados en Matlab.

Método	Medicion de no gausseanidad	Tiempo Ejecución Uniforme	Doble Exp.
Gradiente	Kurtosis	23.67	22.8
	Negentropia	10.71	8.67
FastICA	Kurtosis	1.07	1.21
	Negentropia	2.35	1.81

TABLA 3.1. Comparación entre los tiempos de ejecución de diferentes métodos

Se puede observar que el tiempo de ejecución de los métodos basados en punto fijo son mucho menores que los de gradiente; sea cual sea la distribución a partir de la cual se generaron los datos (tanto kurtosis o negntropía).

Verificación de superioridad de ICA a PCA para comp. indep.

Características de la Prueba

(a) Puntos de arranque aleatorios

(b) 1000 muestras aleatorias diferentes en cada corrida

(c) Métodos programados en Matlab

(d) Criterio de comparación: E1

Distribución	Método	%Superioridad	Tiempo Ejecución
Uniforme	ICA	68	1.1
	PCA	32	0.01
Doble Exponencial	ICA	55.75	1.04
	PCA	44.25	0.01

TABLA 3.2. Comparación entre ICA y PCA

Tiempos de ejecución y comparación entre los métodos

Generación de datos	Método	Tiempo de ejecución ICA	Tiempo de ejecución PCA	Desempeño E1 ICA	Desempeño E1 PCA
Uniforme continua	Gradiente Kurtosis	21.48	0.05	*1.902*	3.803
	Gradiente Negentropia	0.33	0.05	*0.305*	3.803
	FastICA Kurtosis	0.88	0.05	*0.046*	3.803
	FastICA Negentropía	2.25	0.05	*1.294*	3.803
Doble exponencial continua	Gradiente Kurtosis	23.24	0.05	*****	*2.352*
	Gradiente Negentropia	2.352	0.05	2.722	*2.352*
	FastICA Kurtosis	0.93	0.005	*0.151*	2.352
	FastICA Negentropía	2.31	0.05	3.393	*2.352*
Conjunta de dos normales cont	Gradiente Kurtosis	6.7	0.05	*****	*1.561*
	Gradiente Negentropia	81.35	0.05	*0.7*	1.561
	FastICA Kurtosis	1.59	0.05	2.6	*1.561*
	FastICA Negentropía	2.2	0.05	2.6	*1.561*
Exponencial discreta	Gradiente Kurtosis	21.09	0.05	*2.06*	3.423
	Gradiente Negentropia	34.28	0.05	3.33	3.423
	FastICA Kurtosis	0.88	0.05	*0.131*	3.423
	FastICA Negentropía	1.7	0.05	*0.76*	3.423
Doble Exponencial discreta	Gradiente Kurtosis	0.5	0.05	3.517	*3.406*
	Gradiente Negentropia	1.32	0.05	3.715	*3.406*
	FastICA Kurtosis	0	0.05	*1.875*	3.406
	FastICA Negentropía	0.05	0.05	*2.355*	3.406

TABLA 3.3 Comparaciones entre ICA y PCA.

OBSERVACIONES: En dos dimensiones, visualmente los resultados obtenidos no concuerdan con el índice utilizado para medir el desempeño. En cuestión de tiempo, ICA consume mucho más que PCA. Lo anterior es muy claro, ya que requiere encontrar los componentes principales antes de encontrar los s_i. En forma general, considerando los resultados anteriores se puede ver que el 60% de las veces ICA encuentra mejores componentes que PCA.

COMPARACION EN BASE AL ANGULO

Como se mencionó anteriormente, la manera mas simple de encontrar a los componentes independientes es por medio de rotaciones que nos lleven a maximizar la no gaussenidad, de esta manera, como se conoce a la matriz **A** con la que se realizó la mezcla de los datos, se puede utilizar un criterio basado en el ángulo de dicha matriz con la matriz de los componentes independientes encontrados. Mientras más cercano sea el ángulo a 0, entonces los componentes son mejores aproximaciones a los originales.

Características de la Prueba

(a) Puntos de arranque aleatorios

(b) 1000 muestras aleatorias diferentes en cada corrida

(c) Métodos programados en Matlab

(d) Tipo de comparación: Ángulo

		Solución Encontrada		
Distribución	Método	Errónea	Aproximada	Correcta
Uniforme	PCA	80%	15%	5%
	ICA	*	1%	99%
Doble Exp.	PCA	25%	55%	20%
	ICA	*	5%	95%

TABLA 3.4 Comparaciones entre ICA y PCA por criterio del ángulo.

OBSERVACIONES:Se toma como una solución buena aproximada el encontrar unos vectores cuyo ángulo con los esperados sea menor de 8^o de desviación y como correcta aquellos vectores que coinciden exactamente con la matriz que se utilizó para realizar la rotación.

Sensibilidad del punto de arranque

El método de FastICA no es tan sensible al punto de arranque, como el método del gradiente lo es. Esto se debe a que el punto inicial en el método del gradiente determina la dirección hacia donde se va a mover; e influye notablemente en aquellas distribuciones que contienen varios maximos (locales o globales).

Utilizando estas pruebas y las realizadas anteriormente, se puede ver que en el caso de distribuciones continuas, en la mayor parte de los casos los métodos de ICA tienen un mejor desempeño.

Existen otros métodos para calcular los componentes independientes que se describen en el resto del capítulo.

3.3 ICA POR ESTIMACIÓN DE LA MAXIMA VEROSIMILITUD

En ésta sección se describirá otra forma de encontrar los componentes independientes, para ésto es necesario formular el problema de la siguiente manera: Teniendo n observaciones $x(1), x(2), \cdots x(n)$ que contienen información de m cantidades $\theta_1, \theta_2, \ldots, \theta_m$ desconocidas que se desean calcular; el objetivo es encontrar el vector de parámetros $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)$. Una manera es obteniendo el estimador de máxima verosimilitud, que implica encontrar valores de parámetros con los que se obtiene la probabilidad más alta para las observaciones; éste método es descrito a continuación.

3.3.1 EL METODO DE MAXIMA VEROSIMILITUD

Este método supone que los parámetros desconocidos son constantes y que no se conoce información a priori de ellos; de ésta manera el estimador de máxima verosimilitud $\hat{\theta}_{ML}$ del vector $\boldsymbol{\theta}$ es el valor que maximiza a la función de verosimilitud (que es la distribución conjunta)

$$p\left(\mathbf{x}_T|\boldsymbol{\theta}\right) = p\left(x(1), x(2), ..., x(n)|\boldsymbol{\theta}\right)$$

de las observaciones $x(1), x(2), ..., x(n)$.

Como por lo general las funciones de densidad contienen funciones exponenciales, es mas conveniente tomar la función de log verosimilitud $\ln p(\mathbf{x}_T|\boldsymbol{\theta})$, ya que el EMV también la maximiza y se encuentra solucionando la ecuación:

$$\frac{\delta}{\delta\boldsymbol{\theta}} \ln p(\mathbf{x}_T|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{ML}} = \mathbf{0}$$

La construcción de la función de verosimilitud puede ser muy complicada si las medidas dependen de otras, por lo que se supone que el método se aplica a observaciones $x(j)$ que son estadísticamente independientes una de la otra. Asumiendo independencia, la función de verosimilitud es el producto:

$$p(\mathbf{x}_T|\boldsymbol{\theta}) = \prod_{j=1}^{T} p(x(j)|\boldsymbol{\theta})$$

donde $p(x(j)|\boldsymbol{\theta})$ es la función de densidad de probabiliadad condicional de una medida escalar $x(j)$. Si se toma el logaritmo, esto es,

$$p(\mathbf{x}_T|\boldsymbol{\theta}) = \sum_{j=1}^{T} \ln p(x(j)|\boldsymbol{\theta}).$$

El vector de la ecuación de verosimilitud consiste en m ecuaciones escalares

$$\frac{\delta}{\delta\boldsymbol{\theta}_i} \ln p(\mathbf{x}_T|\hat{\boldsymbol{\theta}}_{ML})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{ML}} = 0, i = 1, ..., m$$

Para aplicar el EMV es necesario calcular primero la función de verosimilitud, por lo que se desarrolla en la siguiente sección.

3.3.2 LA VEROSIMILITUD DEL MODELO ICA

Antes de calcular la función de verosimilitud, se mencionará un teorema necesario:

Theorem 1 *La densidad de una transformación lineal $y = g(x)$, donde $x = g^{-1}(y)$ es:*

$$p_y(y) = \frac{p_x(g^{-1}(y))}{|\det Jg(g^{-1}(y))|}$$

donde Jg es el jacobiano

Así, teniendo el vector de mezclas

$\mathbf{x} = \mathbf{As}$

partiendo del teorema anterior se puede formular la densidad p_x como:

$\mathbf{x} = g(\mathbf{s})$

$\mathbf{s} = g^{-1}(\mathbf{x})$

$$p_x(\mathbf{x}) = \frac{p_s(g^{-1}(\mathbf{x}))}{|\det Jg(g^{-1}(\mathbf{x}))|}$$

$$p_x(\mathbf{x}) = \frac{p_s(A^{-1}(\mathbf{x}))}{|\det Jg(A^{-1}(\mathbf{x}))|}$$

como $\mathbf{A}^{-1}= \mathbf{B}$, y $\mathbf{A}^{-1}\mathbf{x} = \mathbf{s}$

$$p_x(\mathbf{x}) = \frac{p_s(\mathbf{s})}{|\det Jg(\mathbf{s})|}$$

por lo tanto

$$p_x(\mathbf{x}) = |\det \mathbf{B}|\, p_s(\mathbf{s}) = |\det \mathbf{B}| \prod_i p_i(s_i)$$

donde p_i denota las densidades de los componentes independientes, mostrándolo como una función de $\mathbf{B}$

$$p_x(\mathbf{x}) = |\det \mathbf{B}| \prod_i p_i(\mathbf{b}_i^T x)$$

donde $\mathbf{B} = (\mathbf{b}_1, ..., \mathbf{b}_n)^T$. Asumiendo que se tienen n observaciones de $\mathbf{x}$, denotadas por $\mathbf{x}(1), \mathbf{x}(2), ...\mathbf{x}(n)$, la verosimilitud puede ser calculada como el producto de las densidades evaluadas en los n puntos.

$$L(\mathbf{B}) = \prod_{t=1}^{n} \prod_{i=1}^{n} p_i(\mathbf{b}_i^T \mathbf{x}(t)) \, |\det \mathbf{B}|$$

por lo que la log verosimilitud es:

$$\log L(\mathbf{B}) = \sum_{t=1}^{n} \sum_{i=1}^{n} \log p_i(\mathbf{b}_i^T \mathbf{x}(t)) + n \log |\det \mathbf{B}|$$

Cambiando notación, podemos denotar la suma sobre el indice t por una esperanza y dividir entre T, obteniendo

$$\frac{1}{N} \log L(\mathbf{B}) = E\left\{ \sum_{i=1}^{n} \log p_i(\mathbf{b}_i^T \mathbf{x}) \right\} + \log |\det \mathbf{B}| \tag{3.32}$$

Una vez que se tiene la expresión de la verosimilitud, todavia faltan conocer las densidades, lo cual es un problema no paramétrico, es decir, que no puede ser reducido a un número finito de parámetros. Debido a esto, a la estimación del modelo de ICA se le conoce como una estimación semiparamétrica.

3.3.3 ESTIMACION DE LA DENSIDAD

Para resolver el problema semiparamétrico anterior se pueden tomar dos caminos: En el primero de ellos, si se conocen las densidades (o aproximaciones a ellas), se pueden utilizar como conocimiento a priori en el modelo. En el segundo caso, que no son conocidas, se pueden aproximar por una familia de densidades que son especificadas por un número limitado de parámetros. Si el número de parámetros necesita ser muy grande, entonces no se gana mucho con ésta aproximación.

Theorem 2 *Denotando p_i como las densidades de los componentes independientes, y $g_i(s_i) = \frac{\delta}{\delta s_i}\log\tilde{p}_i(s_i) = \frac{\tilde{p}_i'(s_i)}{\tilde{p}_i(s_i)}$ como restricciones para la estimación de los componentes independientes $y_i = \mathbf{b}_i^T\mathbf{x}$, de tal manera que estén no correlacionados y con varianza unitaria; entonces, el estimador de MV es localmente consistente, lo cual implica que si se tenía un máximo en un punto, se seguirá teniendo si se asumen densidades $\tilde{p}_i$ tales que $E\{s_i g_i(s_i) - g_i'(s_i)\} \succ 0$, para toda i.*

Basándose en el teorema anterior, se puede demostrar que es suficiente aproximar una densidad por medio de dos funciones (que pueden ser muy simples) [1]. Así que, para cada componente independiente, se tiene que determinar cual de las dos aproximaciones es mejor.

Una propuesta de funciones son las siguientes:

$\log\tilde{p}_i^+(s) = \alpha_1 - 2\log\cosh(s)$

$\log\tilde{p}_i^-(s) = \alpha_2 - [s^2/2 - \log\cosh(s)]$

la razón principal de esto es que una es la negativa de la otra. Se puede observar también que $\tilde{p}_i^+$ es una densidad supergauseana, y $\tilde{p}_i^-$ es subgausseana.El cálculo de los momentos no polinomiales de estas funciones es:

Para $\tilde{p}_i^+ \Rightarrow$

$$\begin{aligned} g_i(s_i) &= \frac{\delta}{\delta s_i}[-2\log\cosh(s)] \\ &= -2\frac{\delta}{\delta s_i}[\log\cosh(s)] \\ &= -2\frac{1}{\cosh(s)}\frac{\delta}{\delta s_i} - (\cosh(s)) \end{aligned}$$

$$\begin{aligned} &= -2\frac{1}{\cosh(s)}senh(s) \\ &= -2\tanh(s) \end{aligned}$$

Para $\tilde{p}_i^- \Rightarrow$

$$\begin{aligned} g_i(s_i) &= \frac{\delta}{\delta s_i} - [s^2/2 - \log\cosh(s)] \\ &= E\{\tanh(s_i)s_i - (1-\tanh(s_i)^2\} \end{aligned}$$

Entonces, se pueden calcular los momentos polinomiales de las dos distribuciones y escoger la que completa la condición de estabilidad. Esto puede realizarse en línea o durante la maximización de la verosimilitud. Este método siempre proporciona un estimador consistente (localmente) y resuelve el problema de estimación semiparamétrica.

A continuación se mencionan los métodos para calcular los componentes independientes por medio de la estimación de máxima verosimilitud

3.3.4 ALGORITMO DE BELL-SEJNOWSKI (METODO DE GRADIENTE)

El método más simple para maximizar la verosimilitud es derivando la ecuación (3.32), donde se tiene

$$\begin{aligned} \frac{1}{T}\frac{\delta\log L(\mathbf{B})}{\delta\mathbf{B}} &= \frac{\delta}{\delta\mathbf{B}}E\left\{\sum_{i=1}^{n}\log p_i(\mathbf{b}_i^T\mathbf{x})\right\} + \frac{\delta\log|\det\mathbf{B}|}{\delta\mathbf{B}} \\ &= E\left\{\frac{\delta}{\delta\mathbf{B}}\sum_{i=1}^{n}\log p_i(\mathbf{b}_i^T\mathbf{x})\right\} + \frac{\delta\log|\det\mathbf{B}|}{\delta\mathbf{B}} \end{aligned}$$

basándonos en la ecuación 9.1, se tiene

$$\frac{1}{T}\frac{\delta\log L(\mathbf{B})}{\delta\mathbf{B}} = E\left\{\mathbf{g}(\mathbf{B}\mathbf{x})\mathbf{x}^T\right\} + \left[\mathbf{B}^T\right]^{-1}$$

donde $\mathbf{g}(\mathbf{y}) = (g_i(y_i), ..., g_n(y_n))$, con $g_i = (\log p_i)' = \frac{p_i'}{p_i}$; por lo que un algoritmo de estimación ML, sería

$$\Delta \mathbf{B} \propto E\left\{\mathbf{g}(\mathbf{Bx})\mathbf{x}^T\right\} + \left[\mathbf{B}^T\right]^{-1} \tag{3.33}$$

puede utilizarse una versión estocástica del algoritmo, esto es, omitiendo el operador de esperanza

$$\Delta \mathbf{B} \propto \mathbf{g}(\mathbf{Bx})\mathbf{x}^T + \left[\mathbf{B}^T\right]^{-1}$$

El algoritmo converge muy lentamente, especialmente por la inversión de la matriz $\mathbf{B}$ necesaria en cada paso.

3.3.5 ALGORITMO DE GRADIENTE NATURAL

El método de gradiente natural fue propuesto por Amari[1] y simplifica la maximización de la verosimilitud considerablemente, e incluso hace que esté mejor condicionado. A grandes razgos, para obtener éste algoritmo se parte del algoritmo del gradiente:

$\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\delta \mathcal{J}(\mathbf{w})}{\delta \mathbf{w}}$ y se modifica de la siguiente manera: $\frac{\delta \mathcal{J}(\mathbf{w})}{\delta \mathbf{w}} = \frac{\delta \mathcal{J}(\mathbf{w})}{\delta \mathbf{w}} \mathbf{W}^T\mathbf{W}$.

Multiplicando el lado derecho de 3.33 por $\mathbf{B}^T\mathbf{B}$, se tiene:

$\Delta \mathbf{B} \propto \left[\mathbf{B}^T\right]^{-1}\mathbf{B}^T\mathbf{B} + E\left\{\mathbf{g}(\mathbf{Bx})\mathbf{x}^T\right\}\mathbf{B}^T\mathbf{B} = \mathbf{B} + E\left\{\mathbf{g}(\mathbf{Bx})\mathbf{x}^T\mathbf{B}^T\right\}\mathbf{B}$,

si $\mathbf{y} = \mathbf{Bx}$

$\Delta \mathbf{B} \propto \mathbf{B} + E\left\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\right\}\mathbf{B} = \left(\mathbf{I} + E\left\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\right\}\right)\mathbf{B}$

Para componentes independientes supergauseanos, la función $g^+(y) = -2\tanh(y)$ y para componentes subgauseanos $g^-(y) = \tanh(y) - y$.

3.3.6 ALGORITMO DE PUNTO FIJO

La verosimilitud puede ser calculada por medio de un algoritmo de punto fijo, tan confiable como el de negentropia. Al igual que en éste, el cual optimizaba $E\{G(\mathbf{w}^T\mathbf{z})\}$, se tiene el mismo problema, ya que bajo la restricción de que los componentes independientes que se encuentren deben de ser blancos el término $\log|\det \mathbf{W}|$ de la ecuación 3.32 es una constante, por lo que se puede utilizar la misma clase de derivación de punto fijo que la utilizada anteriormente.

Asi, se tenia un algoritmo de FastICA para datos blancos:

$\mathbf{w} \leftarrow \mathbf{w} - \frac{E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\} + \beta\mathbf{w}}{E\{g'(\mathbf{w}^T\mathbf{z})\} + \beta}$

donde $\beta = -E\{y_i g(y_i)\}$. Escribiendo la ecuación de manera matricial se tiene:

$\mathbf{W} \leftarrow \mathbf{W} + diag(\alpha_i)\left[diag(\beta_i) + E\left\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\right\}\right]\mathbf{W}$

donde $\alpha_i = -1/\left(E\left\{g'(\mathbf{w}^T\mathbf{z})\right\} + \beta_i\right)$, y $\mathbf{y} = \mathbf{Wz}$. Expresandolo para datos no blancos, se reemplaza $\mathbf{W}$ por $\mathbf{Wz} = \mathbf{WVx}$ que es $\mathbf{B} = \mathbf{WV}$, obteniéndose

$\mathbf{B} \leftarrow \mathbf{B} + diag(\alpha_i)\left[diag(\beta_i) + E\left\{\mathbf{g}(\mathbf{y})\mathbf{y}^T\right\}\right]\mathbf{B}$

Después de cada iteración, la matriz $\mathbf{B}$ se proyecta sobre el conjunto de matrices blanqueadas. Ésto se puede realizar con $\mathbf{B} \leftarrow \left(\mathbf{BCB}^T\right)^{-1/2}\mathbf{B}$ donde $\mathbf{C} = E\{\mathbf{xx}^T\}$

3.4 ICA POR MINIMIZACIÓN DE LA INFORMACIÓN MUTUA

La información mutua es una medida de la dependencia entre variables aleatorias. Ésta es siempre no negativa y cero solo en el caso de variables estadísticamente independientes. La información mutua considera la estructura de las variables y no lo solamente la covarianza entre ellas como PCA.

Así, la información mutua se puede utilizar para encontrar los componentes independientes. Supongamos el modelo de ICA como un vector $\mathbf{x}$ con una transformación invertible $\mathbf{B}$, de tal manera que $\mathbf{B}$ proporcione la minimización de la información mutua de los componentes $s_i(\mathbf{s} = \mathbf{Bx})$. De esta manera la minimización de la información mutua puede ser interpretada como la maximización de los componentes independientes.

La información mutua I entre m variables aleatorias, $y_i, i = 1, ..m$ está definida como:

$I(y_1, y_2, ..., y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y})$

donde H es la entropia diferencial $H(\mathbf{y}) = -\int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y}$

y una versión normalizada de ella es la negentropia $J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$

En general, se puede demostrar que $I(y_1, y_2, ..., y_m) = const - \sum_i J(y_i)$ cuando los datos están decorrelacionados [1].

Asi, se puede ver que la estimación de los componentes independientes por la minimización de la información mutua es equivalente a la maximización de la suma de no gauseanidades de los componentes estimados, cuando están decorrelacionados.

Los algoritmos utilizados son los mismos que cuando se calcula la no gauseanidad o los de

estimación de máxima verosimilitud.

3.5 COMPARACIONES ENTRE ALGORITMOS

Como se mencionaron anteriormente, existen varios métodos para calcular los componentes independientes dado un conjunto de datos, la información mutua da un punto de comparación entre los diferentes principios, sabemos que

$$I(y_1, y_2, ..., y_n) = \sum_i H(y_i) - H(\mathbf{x}) - \log|\det \mathbf{B}|$$

bajo la restricción de que y_i son no correlacionados y con varianza unitaria, el ultimo término es una constante y el segundo no depende de $\mathbf{B}$; por lo que se puede decir que la entropía está maximizada por una distribución gauseana cuando la varianza es una constante. Así pues, la minimización de la información mutua es similar a la maximización de la suma de las no gausseanidades de los componentes estimados.

También la información mutua puede ser calculada aproximando las densidades de los ICs por alguna familia paramétrica y utilizando la log-densidad en la definición de entropia, lo que es equivalente a la estimación de máxima verosimilitud.

3.5.1 DIFERENCIAS ENTRE LOS PRINCIPIOS DE ESTIMACIÓN

Algunas diferencias entre los algoritmos son las siguientes:

1. Algunos principios (especialmente la maximización de la no gausseanidad) son mejores para calcular cada componente a la vez, mientras que otros están diseñados para calcular todos los componentes independientes al mismo tiempo (por ejemplo utilizando información mutua, no se puede calcular uno por uno).

2. Algunas funciones objetivo utilizan funciones polinomiales basadas en pdfs, mientras que otros las basan en acumulativas [1]

3. En varios principios de estimación, los ICs que se obtienen están restringidos a ser no correlacionados.

4. Una diferencia práctica importante es que en las estimaciones ML las densidades de los componentes son fijas; ya que las pdfs de los CI no son requeridas con gran presición (solo

necesita saberse si es subgausseana o supergausseana), pero si la naturaleza de los componentes no es correcta, provoca grandes fallas. En contraste, métodos como el de negentropia no tienen éste tipo de problemas[1].

5. Una diferencia entre los algoritmos que utilizan la no gausseanidad es que forzan a los componentes independientes a ser no correlacionados. Ésto no necesariamente se cumple cuando se utiliza información mutua.

Chapter 4

RELACIONES ENTRE ICA Y OTROS METODOS

El objetivo de la primera sección es presentar algunas técnicas muy relacionadas con el Análisis de Componentes Independientes, sin utilizar la formulación estricta de variables latentes, lo cual se hará en la segunda parte.

4.1 COMPONENTES PRINCIPALES

El análisis de componentes principales (PCA) es posiblemente la técnica de reducción de dimensión más utilizada en práctica, por su sencillez y los eficientes algoritmos que existen para calcularlos. En procesamiento de señales, a ésta técnica se le conoce como la transformada de Karhunen-Loeve.

Considerando un vector de variables aleatorias $\{\mathbf{x}_i\}_{i=1}^N$ en $\mathbb{R}^N$ con media $\bar{\mathbf{x}} = (E(\mathbf{x}_1), \ldots, E(\mathbf{x}_N))$ y matriz de covarianza $\sum = E\left\{(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\right\}$, con descomposición $\sum = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ ($\mathbf{U}$ ortogonal y $\mathbf{\Lambda}$ diagonal).

La transformación de componentes principales $\mathbf{y} = \mathbf{U}^T(\mathbf{x} - \bar{\mathbf{x}})$ hace referencia a un sistema en el que la muestra tiene media $\mathbf{0}$ y matriz de covarianza $\mathbf{\Lambda}$ (que contiene los eigenvalores de $\sum$). Se pueden descartar las variables que tienen poca varianza, y proyectar los datos en el subespacio generado por los L principales componentes principales, obteniendose una buena aproximación (fig.4.1).

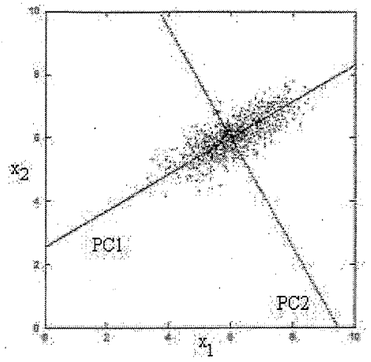


Figure 4-1: Utilización de los primeros componentes principales para un conjunto de datos

Geométricamente, el hiperplano generado por los primeros L componentes principales es el hiperplano de regresión que minimiza la distancia ortogonal entre los datos (fig. 4.1) y comunmente se les utiliza como puntos de arranque de otros algoritmos, como projection pursuit, que se describirá a continuación.

4.2 PROJECTION PURSUIT

Projection pursuit es una técnica desarrollada en estadística para encontrar proyecciones "interesantes" en datos multidimensionales. Éstas proyecciones pueden ser utilizadas para visualizaciones óptimas de los datos.

Cuando projection pursuit es utilizada para análisis exploratorio de datos, comúnmente se calculan las proyecciones mas interesantes en 1D. Este método es una extensión del clásico método de PCA para visualización; el cual presenta la distribución de los datos en el plano formado por los dos primeros componentes principales. La pregunta básica en projection pursuit es definir cuál o cuales son las proyecciones interesantes. Para PCA, las distribuciones mas "interesantes" son aquellas que tienen mayor variabilidad en los datos proyectados. Algunos métodos de análisis multivariado clásico son casos especiales de ésta técnica.

Las desventajas de éste método(y de todos en general) son que:

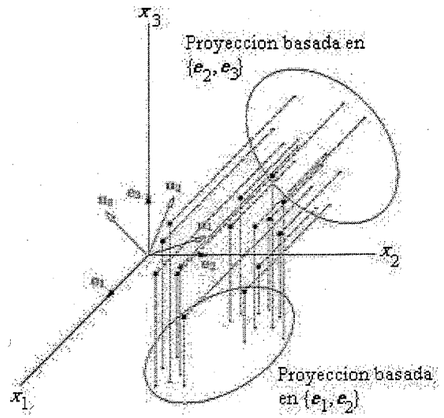


Figure 4-2: Proyecciones "interesantes"

1. Trabaja con proyecciones lineales y por consiguiente no son buenas con estructuras no lineales.

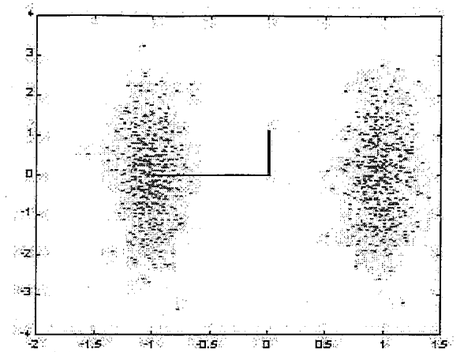
2. Tienden a ser computacionalmente intensivos.

Se considera que una proyección es interesante si presenta una estructura lineal (las correlaciones entre las variables son detectadas por una regresión lineal) o bien, una no lineal (clustering, kurtosis, sezgamiento, etc). Por ejemplo, en la fig.4.2, la proyección en $\{\mathbf{e}_2, \mathbf{e}_3\}$ no da información clara, mientras la de $\{\mathbf{e}_1, \mathbf{e}_2\}$ presenta clusters claros, por lo que la proyección interesante es la segunda.

En la fig.4.3, se tiene otro ejemplo; en este caso, la proyección menos gausseana da una mejor estructura de clusters y la dirección encontrada por PCA no dice nada. Esto muestra como PCA no devuelve siempre una estructura en clusters, ya que esta no es visible en la matriz de varianza y de correlación en la que se basa PCA.

4.3 ANALISIS DE FACTORES

Un método muy relacionado con PCA es el análisis de factores, aquí se utiliza el siguiente modelo:



Primer vector que encuentra ICA (linea mas clara), primer vector que encuentra PCA (linea mas oscura)

Figure 4-3: Comparación entre los primeros vectores encontrados con FastICA y con PCA.

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \tag{4.1}$$

donde $\mathbf{x}$ es el vector de variables observadas, $\mathbf{s}$ el de vector de factores, $\mathbf{A}$ es una matriz de mxn, y el vector $\mathbf{n}$ es el ruido, de la misma dimensión, m, que $\mathbf{x}$. Las variables en $\mathbf{s}$ y $\mathbf{n}$ se asumen como gauseanas donde $\mathbf{s}$ tiene menor dimensión que $\mathbf{x}$. Una manera para estimar el modelo se basa en los factores principales. La idea es aplicar PCA a los datos $\mathbf{x}$ de una manera tal que el ruido sea tomado en cuenta. La forma más simple es asumir una que la matriz de covarianza del ruido es conocida $\mathbf{\Sigma} = E\{\mathbf{n}\mathbf{n}^T\}$, después encontrar los factores aplicando PCA sobre la matriz de covarianza modificada $\mathbf{C} - \mathbf{\Sigma}$ donde $\mathbf{C}$ es la matriz de covarianza de $\mathbf{x}$. Entonces el vector $\mathbf{s}$ es simplemente el vector de componentes principales sin ruido.

La diferencia principal entre FA y PCA es que en FA se busca una rotación de los factores para que se obtenga una base con propiedades interesantes, esto es posible ya que la ecuación (4.1), no proporciona factores únicos.

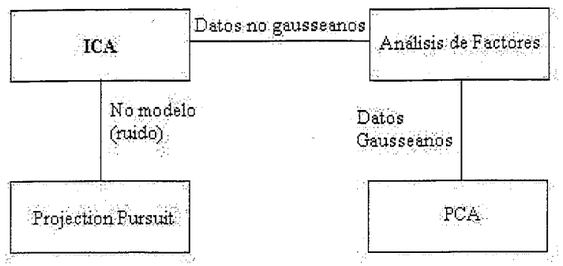


Figure 4-4: Relaciones entre ICA y otros métodos. Las líneas muestran conexiones cercanas.

4.4 RELACIONES DE ICA CON LOS MÉTODOS ANTERIORES

ICA está cercanamente relacionada con los métodos expuestos anteriormente, en las siguientes maneras:

1. En el modelo libre de ruido, la estimación del modelo de ICA significa que se buscan direcciones "interesantes", las cuales dan estimaciones de los componentes independientes. Entonces, podría decirse que ICA es un caso especial de projection pursuit.

2. Comparando el modelo con ruido de ICA con el modelo de análisis de factores, se puede ver que ICA podria ser considerado un análisis de factores no gauseanos. La principal diferencia es que en ICA la reducción de dimensionalidad no es necesariamente la meta.

3. La relación entre análisis de componentes principales es que, ambos métodos formulan una función de objetivo general y luego la maximizan. También, ambos métodos están relacionados con el Análisis de Factores (uno para datos no gausseanos y el otro para gausseanos). La diferencia principal es que PCA utiliza estadísticas de segundo orden, mientras ICA utiliza un orden mayor.

Las líneas en el diagrama 4.4 muestran las relaciones cercanas, primero, si no se supone nada en los datos, ICA es considerado un método de projection pursuit. Si se asume un modelo de

datos con ruido, entonces ICA es considerado una variación de análisis de factores para datos no gausseanos. La conexión entre ICA y PCA es considerada indirecta, ya que se necesita análisis de factores para datos gauseanos[6].

4.5 PANORAMA DE ICA POR MEDIO DE MODELACIÓN CON VARIABLES LATENTES CONTINUAS

En éste sección, se presenta una introducción a los modelos de variables latentes continuas. Las definiciones se dan de una manera superficial, ya que no es el objetivo de la tesis. La meta es mostrar en perspectiva en donde se encuentra ICA dentro de los demás modelos semejantes. Para ahondar más en el topico, referirse a [3].

Los modelos de variables latentes continuas, son en general un tipo de modelos probabilísticos que tratan de explicar procesos en alta dimensionalidad en términos de un menor número de variables. Estos tipos de modelos se utilizan principalmente para resolver problemas de reconocimiento de patrones, como la reconstrucción de datos secuenciales (tratar de recuperar la señal original, dada una secuencia de vectores donde algunos valores están perdidos). La forma [3] en la que se describen los modelos de variables latentes es:

1. Formulación del modelo.
2. Estimación de parámetros (por medio de máxima verosimilitud)
3. Pruebas de hipótesis acerca de los parámetros.

4.5.1 DEFINICIÓN DE VARIABLES LATENTES

Considerando un sistema gobernado por L variables independientes $x_1, x_2, ...x_L$, con las cuales se representan vectores L-dimensionales $\mathbf{x} = (x_1, ..., x_L)^T$ que toman valores en un subconjunto $\mathcal{S}$ de $\mathbb{R}^L$, llamado *espacio latente o de estados.* Es decir cada estado del sistema está dado por un vector $\mathbf{x}$ en particular en el espacio $\mathcal{S}$. La dimensión L del sistema no es conocida por el observador que diseña el experimento con el que se obtienen los datos; pero usualmente el número de variables D es más grande que el número de variables L. Cualquier variable medida puede ser representada como la operación de obtenerla del espacio latente $\mathbf{f} : \mathcal{S} \subset \mathbb{R}^L \rightarrow \mathcal{M} \subset \mathbb{R}^D$. Se asume que $\mathbf{f}$ es no singular (dimensión de la imagen $\mathbf{f}$ es igual que la dimensión de su

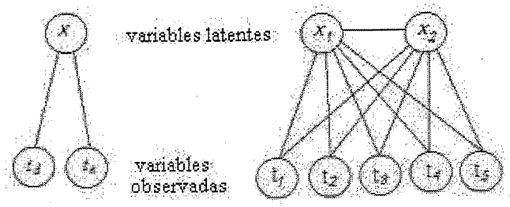


Figure 4-5:

dominio; $\dim \mathcal{S} = \dim \mathbf{f}(\mathcal{S})$), por lo que las variables observadas pertenecen a un espacio lineal L-dimensional $\mathcal{M} = \mathbf{f}(\mathcal{S})$ en $\mathbb{R}^D$ donde $\mathbb{R}^D$ es conocido como el *espacio de observaciones.* $\mathbf{f}$ describe un proceso ideal, por lo que cada punto del espacio latente es mapeado exactamente a un punto en el espacio $\mathcal{M}$, por lo que se puede invertir. De ésta manera $\mathbf{f}^{-1}$:$\mathcal{M} \in \mathbb{R}^D \rightarrow \mathcal{S} \in \mathbb{R}^L$, pero para cada punto $\mathbf{x}$ en el espacio latente, el observador tiene $\mathbf{f}(\mathbf{x}) + \mathbf{e}$ en el espacio de datos, donde $\mathbf{e}$ es el error estocástico (ruido), $\mathbf{e} \in \mathbb{R}^D$. El modelo de ruido se representa como una función de densidad $p(\mathbf{t}|\mathbf{x})$, que representa la probabilidad de que el punto latente $\mathbf{x}$ sea observado como el punto $\mathbf{t}$. El ruido generalmente se asume como $\mathcal{N}(\mathbf{f}(\mathbf{x}), \sum)$.

4.5.2 MODELO GENERATIVO USANDO VARIABLES LATENTES CONTINUAS

En la modelación de variables latentes, se asume que los datos en alta dimensión son generados por un proceso en dimensión más baja, donde el objetivo es aprender de ellas y aplicarlas en dimensiones más altas (modelo generativo). Las variables latentes están mapeadas por una transformación fija (procedimiento de medida) y posteriormente se les añade ruido (variación estocástica). Asumiendo que las variables latentes y las observadas son continuas, llamemos $\mathcal{T} \subseteq \mathbb{R}^D$ al espacio observado o de datos D-dimensional. Considerando una distribución desconocida $p(\mathbf{t})$ en el espacio de datos para $\mathbf{t} \in \mathcal{T}$, del cual únicamente tenemos una muestra $\{\mathbf{t}_n\}_{n=1}^N \subset \mathcal{T}$. Como $L < D$, al espacio L-dimensional se le llamaremos espacio latente $\mathcal{X} \subseteq \mathbb{R}^L$.

Un punto $\mathbf{x}$, en el espacio latente $\mathcal{X}$ es generado de acuerdo a una distribución a priori $p(\mathbf{x})$ y se mapea a un espacio de datos T por medio de un mapeo suave $\mathbf{f} : \mathcal{X} \rightarrow \mathcal{T}$.se obtiene un

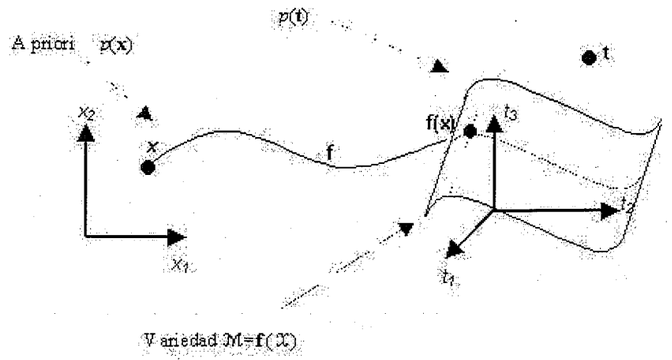


Figure 4-6:

modelo de error o de ruido $p(\mathbf{t}|\mathbf{x}) = p(\mathbf{t}|\mathbf{f}(\mathbf{x}))$. La función de densidad de probabilidad en el espacio $\mathcal{T}$x$\mathcal{X}$ es $p(\mathbf{t}, \mathbf{x})$ e integrando sobre el espacio latente se obtiene la ecuación fundamental de modelos de variables latentes.

$$p(\mathbf{t}) = \int_{\mathcal{X}} p(\mathbf{t}, \mathbf{x}) d\mathbf{x} = \int_{\mathcal{X}} p(\mathbf{t}|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \tag{4.2}$$

4.5.3 MODELO DE RUIDO

Si las variables latentes son eficientes para representar a las variables observadas, se espera que dado un valor de latentes, los valores del grupo de observadas sean independientes de los valores de otro grupo. De otra manera, las variables latentes escogidas, no explican necesariamente las correlaciones entre las observadas y las latentes. Entonces, para todas las $d, e \in \{1, ...D\}$,

$$p(t_d|t_e, \mathbf{x}) = p(t_d, \mathbf{x}) \Rightarrow p(t_d, t_e|\mathbf{x}) = p(t_d|t_e, \mathbf{x}) p(t_e|\mathbf{x}) = p(t_d|\mathbf{x}) p(t_e|\mathbf{x}) \tag{4.3}$$

donde, la distribución de las variables observadas condicionadas a las variables latentes, o modelo de ruido, es factorial.

$$p(\mathbf{t}|\mathbf{x}) \stackrel{def}{=} \prod_{d=1}^{D} p_d(t_d|\mathbf{x}) \tag{4.4}$$

Esto es, para $L \leq D$, las variables observadas son independientes condicionalmente dadas las variables latentes. Ésto es usualmente conocido como el axioma de la independencia local (o condicional), éste axioma es una definición de que se puede encontrar la distribución conjunta de las variables observadas en término de las variables latentes. La idea es encontrar el menor número de variables latentes que realicen esto.

Tomando el modelo de ruido $p(\mathbf{t}|\mathbf{f}(\mathbf{x}))$, se puede utilizar una función de densidad con las propiedades[ref]:

1. Centrada en $\mathbf{f}(\mathbf{x})$. Esto es, $\forall \mathbf{x} \in \mathcal{X} : E\{\mathbf{t}|\mathbf{x}\} = \mathbf{f}(\mathbf{x})$

2. Que decaiga gradualmente conforme la distancia de $\mathbf{f}(\mathbf{x})$ se incrementa, de acuerdo a algún parámetro relacionado con la covarianza del ruido.

3. Con densidad no cero para cada punto del espacio observado (para que ninguna región del espacio observado tenga probabilidad nula).

4. Tiene que tener una matriz de covarianza diagonal para contar con diferentes escalas en las diferentes variables observadas.

4.5.4 DISTRIBUCION A PRIORI

Dada cualquier distribución a priori $p_{\mathbf{x}}(\mathbf{x})$ de L variables latentes $\mathbf{x}$, siempre es posible encontrar una transformación invertible $\mathbf{g}$ para un conjunto alternativo de L variables latentes $\mathbf{y} = (y_1, ..., y_L) = \mathbf{g}(\mathbf{x})$ teniendo otra distribución deseada $p_{\mathbf{y}}(\mathbf{y})$:

$$\mathcal{X} \xrightarrow{g} \mathcal{Y} \xrightarrow{f} \mathcal{T} \tag{4.5}$$

Donde el mapeo del nuevo espacio latente al espacio de datos es $f = fog$, $\mathbf{t} = \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{g}^{-1}(\mathbf{y})) = \mathbf{f}(\mathbf{y})$ y la nueva distribución a priori de las variables $\mathbf{y}$ es $p_{\mathbf{y}}(\mathbf{y}) = p_{\mathbf{x}}(\mathbf{x})|\mathbf{J}_g|^{-1}$, donde $\mathbf{J}_g \stackrel{def}{=} \left(\frac{\delta g_1}{\delta x_k}\right)$. Se necesita que el mapeo $\mathbf{f} : \mathcal{X} \to \mathcal{T}$ sea suave (continuo y diferenciable) debido a:

1. Continuidad. Garantiza que los puntos que están cerca de otros en el espacio latente, al ser mapeados, estén cerca del unos a otros en el espacio de datos.

2. Diferenciabilidad. Es un requerimiento práctico para poder utilizar métodos de optimización basados en derivadas.

4.5.5 ESTIMACIÓN DE PARÁMETROS

La distribución a priori en el espacio latente $p(\mathbf{x})$, el mapeo suave $\mathbf{f}$ y el modelo de ruido $p(\mathbf{t}|\mathbf{x})$ están equipados con parámetros conocidos como $\boldsymbol{\Theta}$. Los parámetros tienen que ser optimizados, para maximizar la verosimilitud de los datos observados dados los parámetros, $p(\mathbf{t}_n|\boldsymbol{\Theta})$. Desde un enfoque bayesiano, la distribución a priori es colocada en los parámetros y las inferencias posteriores son realizadas marginalizando los parámetros.

$$p(\mathbf{t}, \mathbf{x}) = \int p(\mathbf{t}, \mathbf{x}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}) d\boldsymbol{\Theta} \tag{4.6}$$

donde $p(\boldsymbol{\Theta})$ es la distribución del parámetro a priori.

La log-verosimilitud de los parámetros dadas las muestras $\{\mathbf{t}_n\}_{n=1}^N$ es:

$$L(\boldsymbol{\Theta}) \stackrel{def}{=} \ln p(\mathbf{t}_1, ...\mathbf{t}_N|\boldsymbol{\Theta}) = \ln \prod_{n=1}^{N} p(\mathbf{t}_n|\boldsymbol{\Theta}) = \sum_{n=1}^{N} \ln p(\mathbf{t}_n|\boldsymbol{\Theta}) \tag{4.7}$$

lo cual implica un conjunto de valores para los parámetros $\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta})$ que corresponde a los máximos locales de la log-verosimilitud. Una estrategia de maximización es el algoritmo EM.

Paso E: Calcula la esperanza de la log-verosimilitud de los datos completos con respecto a la distribución a posteriori.

Paso M: Determina los nuevos valores de los parámetros $\boldsymbol{\Theta}^{\tau+1}$ que maximizan la esperanza de todas las verosimilitudes de los datos.

Las desventajas son que es un algoritmo batch (secuencial) y la convergencia es lenta después de los primeros pasos (que si son efectivos).

4.5.6 MODELOS ESPECÍFICOS DE VARIABLES LATENTES

La tabla 4.7 menciona los diferentes modelos formados a partir de variables latentes.

Un modelo de variables latentes es especificado con los siguientes elementos:

1. La distribución a priori en el espacio latente $p(\mathbf{x})$

Modelo	A priori espacio latente p(x)	Mapeo f x --> t	Modelo de ruido p(t|x)	Densidad en el espacio observado p(t)
Análisis de Factores	N(0,I)	lineal	diagonal	restringida gaussena
(FA)			normal	
Análisis de Componentes	N(0,I)	lineal	esférica	restringida gaussena
Principales (PCA)			normal	
Analisis de Componentes	no conocido,	lineal	delta de	depende
Independientes (ICA)	pero factorizable		Dirac	
Análisis de Factores	producto mezclas	lineal	normal	mezcla restringida
Independientes (IFA)	gausseanas en 1D			gausseana

Figure 4-7:

2. La función de mapeo suave $\mathbf{f} : \mathcal{X} \rightarrow \mathcal{T}$ del espacio latente al espacio de datos.

3. El modelo de ruido en el espacio de datos $p(\mathbf{t}|\mathbf{x})$

Los modelos latentes pueden ser clasificados como *lineales* y *no lineales* de acuerdo a las características de la función de mapeo $\mathbf{f}$. Se le llama a un modelo de variables latentes, normal, cuando la a priori en el espacio latente y el modelo de ruido son normales.

Part II

APLICACIONES

Chapter 5

ICA EN PROCESAMIENTO DE IMAGENES

A partir de éste capítulo se presentan algunas de las propuestas de uso de ICA en diferentes campos, empezando con procesamiento de imágenes. También se mencionan algunas de las líneas de investigación que ya se han explorado, especialmente por el grupo de la Universidad de Helsinki. Por ejemplo una de las aplicaciones de ICA en ésta área se presenta en [1], en donde se utiliza un método conocido como "sparce code shrinkage method", en el que se encuentran los componentes independientes en ventanas de 8x8 pixeles de una imagen real con el objetivo de eliminar el ruido; el modelo de ICA utilizado es el que contine ruido (Apéndice 3). Las siguientes secciones presentan algunas de las aplicaciones propuestas en ésta tesis.

5.1 SEPARAR IMAGENES MEZCLADAS

El primer problema es encontrar las imágenes originales partiendo de varias que son una mezcla lineal de ellas, utilizando el modelo de ICA libre de ruido y considerando cada imágen como un vector. Para ésto se realizaron varias pruebas en donde se manejan imágenes en escalas de grises y en su mayoría reales; la combinación lineal se hizo mediante una matriz **A**, donde cada uno de sus elementos toma valores aleatorios entre 1-4. El rango de las imágenes finales es diferente al de las originales, pero para comparar los resultados, se realizaron los reescalamientos necesarios. En todas las pruebas se presentan las imagenes originales; las que se introdujeron al

algoritmo (se les llama "mezcladas"), los resultados si únicamente se utiliza el procedimiento de blanqueado y finalmente las imágenes que se recuperaron con ICA. Estas pruebas ilustran que tan aplicable es el método de ICA con medición de gausseanidad por kurtosis. A continuación se presentan la descripción de cada una:

1. Se utilizaron dos imágenes en escala de grises de 16x16 pixeles; los resultados se muestran en la figura 5.1. Se observa que el blanqueado no es suficiente para recuperar las imágenes originales, pero al aplicar ICA se ve una mejora considerable aunque aún asi se tiene un error que se muestra en la fig. 5.2.

2. Esta prueba (fig.5.3) consistió en la separación de 3 imágenes sintéticas de diferentes tonalidades; en las imágenes resultantes se observa que en algunos casos se obtienen los negativos de las originales, como en la primera y la tercera. Ésto no es un problema, ya que simplemente es multiplicar la imagen por -1. Al igual que en la prueba anterior, la recuperación es buena aunque contiene varios errores visibles.

3. Utilizando imágenes reales se observan resultados similares (fig.5.4), el mismo comportamiento en el error total (fig.5.5). También se presenta la gráfica de un renglón escogido al azar (fig. 5.6). A partir de ésta prueba se utilizaron tamaños de 128 x 128 pixeles.

4. Realizando la misma prueba en una variación de tono (que se obtuvo utilizando la mezcla de una imágen real y una rampa) se tiene un comportamiento semejante (fig.5.7).

5. No todos los resultados son tan buenos como los anteriores, ya que con el método de FastICA con negentropía y aplicado a imágenes reales (fig.5.8) pueden verse grandes variaciones en los rangos de los errores(fig.5.9).

6. Finalmente, para mostrar los alcances de la técnica se utilizaron tres y cuatro imágenes mezcladas (fig 5.10 y fig. 5.11).

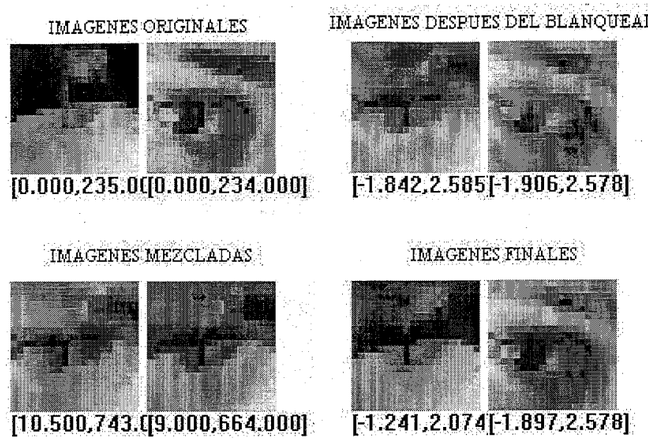


Figure 5-1: Prueba con una imágen (16 x 16 pixeles)

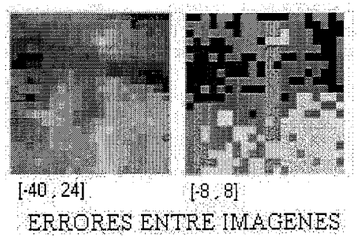


Figure 5-2: Errores entre las imágenes originales y las finales de la fig.5.1

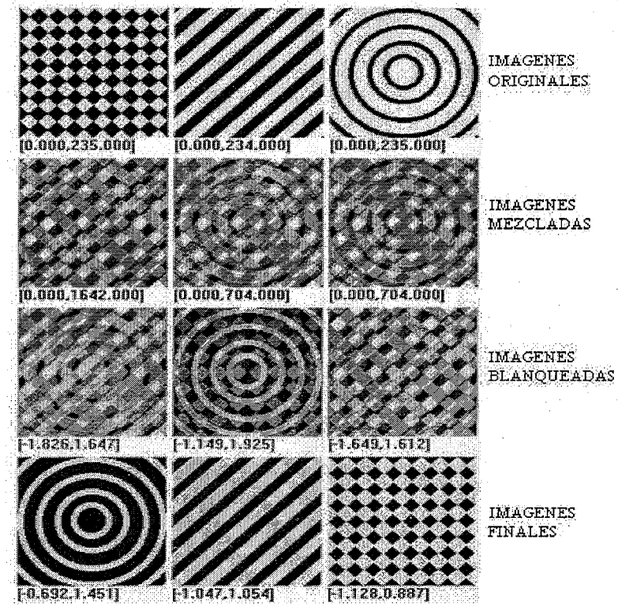


Figure 5-3: Prueba de separación con tres imágenes sintéticas

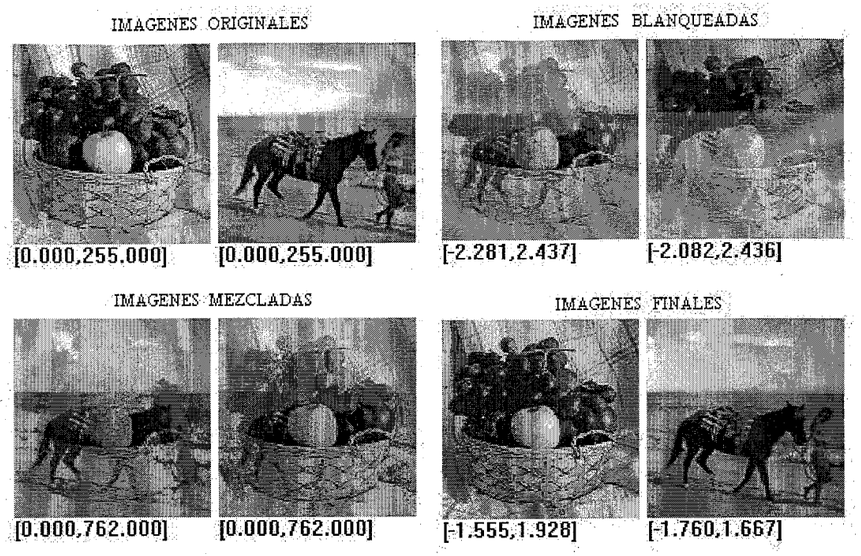


Figure 5-4: Resultados de la separación de dos imágenes reales mezcladas

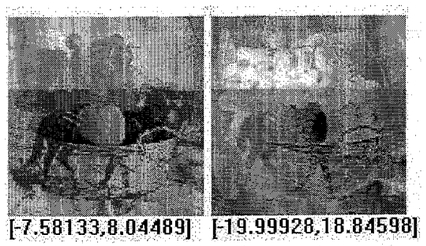


Figure 5-5: Errores entre las imágenes originales y las recuperadas (Fig. 5.4)

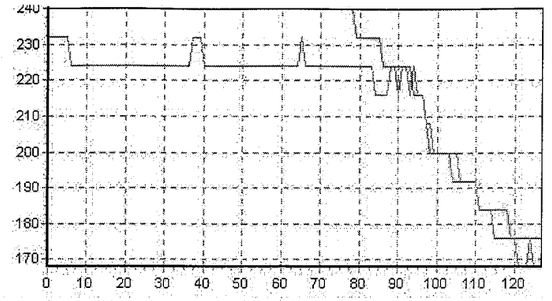


Figure 5-6: Gráfica del renglón 34 de ambas imágenes. Se puede observar el error.

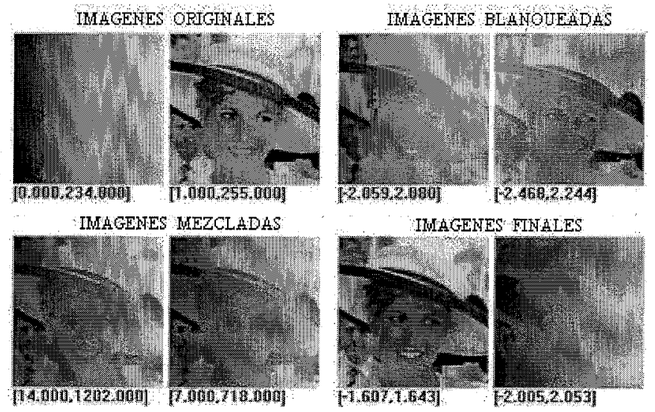


Figure 5-7: Imagen real mezclada con una rampa.

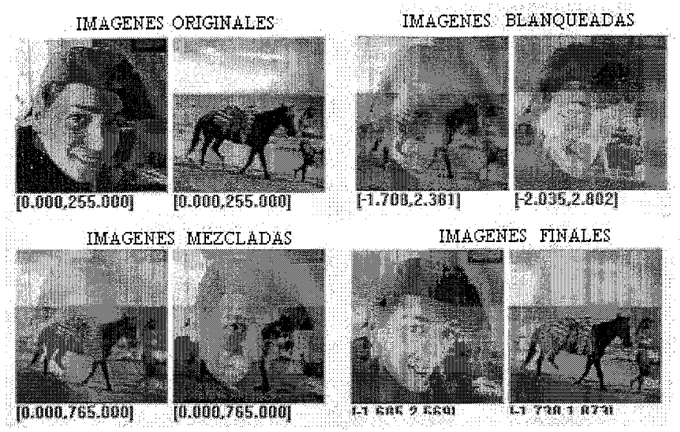


Figure 5-8: Prueba con el método de FastICA (Punto fijo con negentropía).

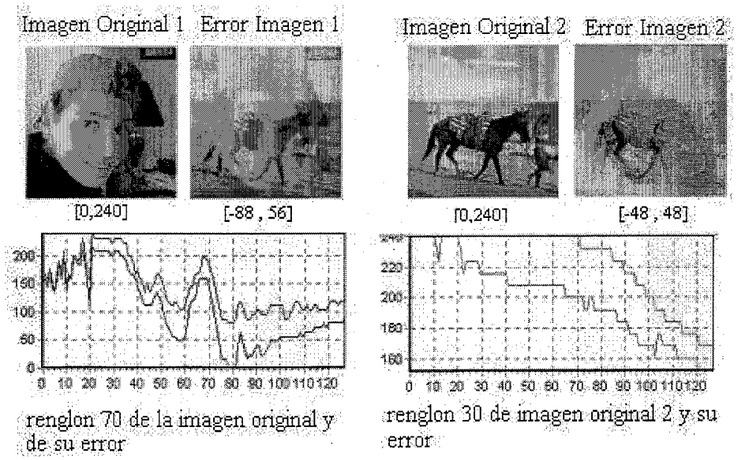


Figure 5-9: Errores de la fig. 5.8

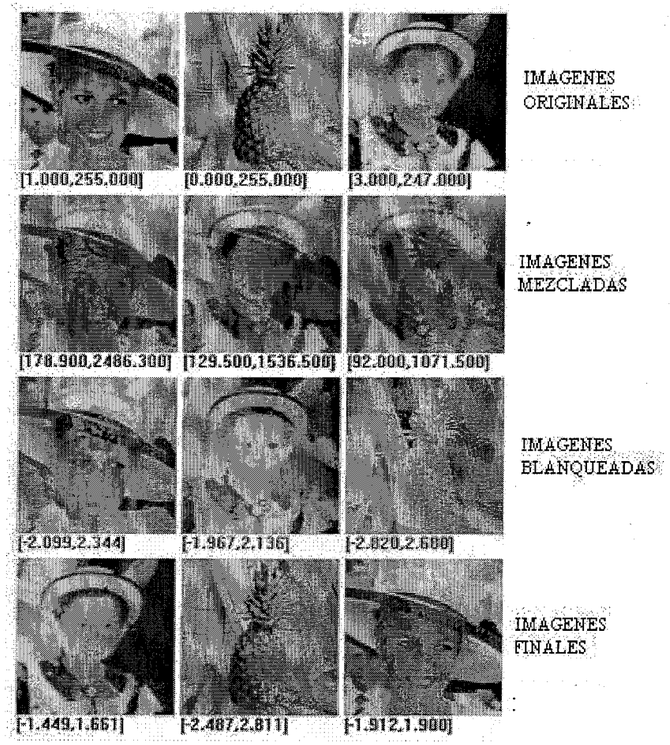


Figure 5-10: Separación de tres imágenes.

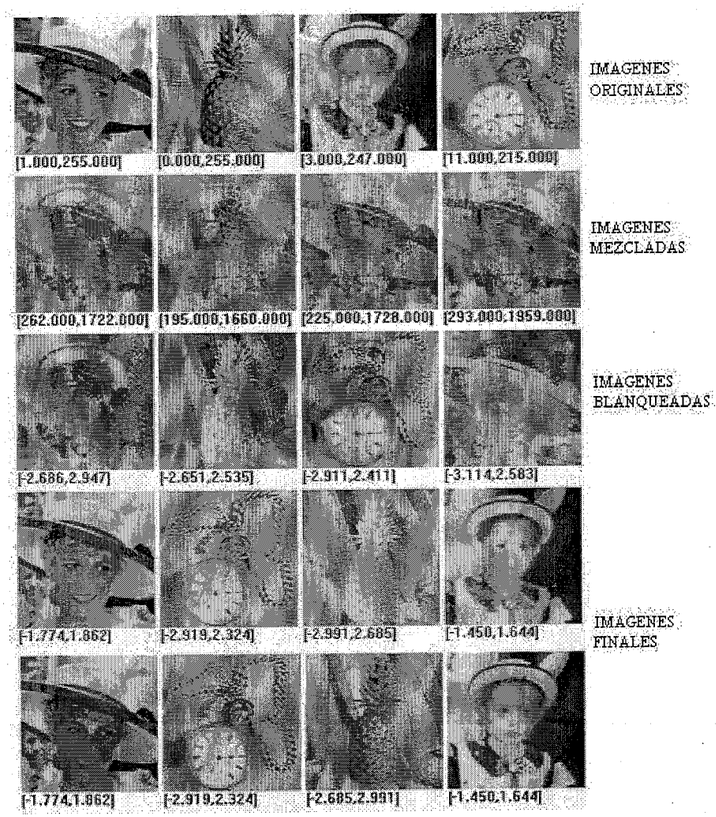


Figure 5-11: Separación de 4 imágenes reales (en la imagen 3 se tuvo que realizar una inversión a la imagen).

5.2 ELIMINACION DE RUIDO

Uno de los principales problemas que se presentan en el manejo de imágenes es la presencia de ruido; en ésta sección se propone una manera de eliminarlo por medio del modelo de ICA libre de ruido. La forma en la que se realizaron las pruebas fue igual que la anterior, pero introduciendo ruido gausseano en una de las imágenes con las que se realizó la mezcla. Anteriormente se habia mencionado que no se iban a utilizar señales con éste tipo de distribución, pero se ve mas adelante que los métodos funcionan adecuadamente con a lo más un componente gausseano. Se partieron de varios casos que se describen a continuación:

1. Tomando como mezcla la misma imagen pero con diferente nivel de ruido (Fig.5.12) se obtienen buenos resultados. Similares si se toma otra combinación lineal de las mismas imágenes(Fig.5.13).

2. No es común que se tengan imágenes del mismo objeto pero con diferentes niveles de ruido, mas bien, se tiene una sola representación con ruido, así que, para eliminarlo se utilizó la misma imagen dos veces y se probó el desempeño del método con diferentes niveles de ruido(Fig.5.14). Se tienen recuperaciones buenas para niveles bajos (que son los más comunes).

3. Un método sencillo de eliminación de ruido es realizar un suavizamiento de la imágen con sigma dependiendo del nivel de ruido; comparándolo con el método de ICA, se observa que el segundo respeta más los bordes y el rango del error es mucho menor. Para ésta prueba se introdujo la imagen ruidosa dos veces al método (fig.5.15) y dos imágenes con diferentes niveles de ruido (fig. 5.16). Para el filtrado clásico, se hizo un promedio de las dos imágenes y sobre el resultado se aplicó el kernel.

4. Comparando el método de ICA con PCA (Fig.5.17)se ve la superioridad de ésta técnica.

Como puede observarse una de las ventajas de ICA para eliminar el ruido es que no pierde los bordes de las imagenes. Así como también se puede apreciar que para éste tipo de aplicaciones el método de ICA presenta un mejor desempeño que PCA.

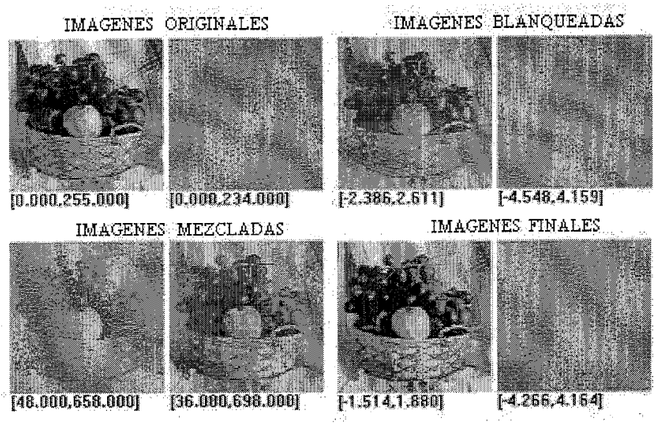


Figure 5-12: Imagen con ruido

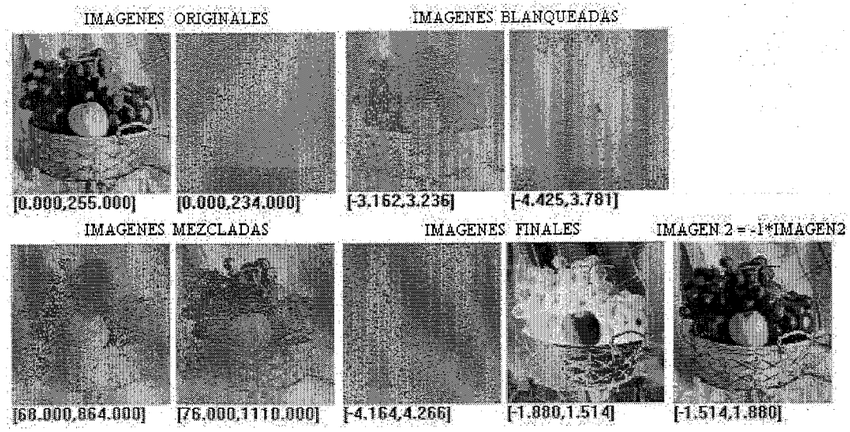


Figure 5-13: Misma imagen con diferente nivel de ruido

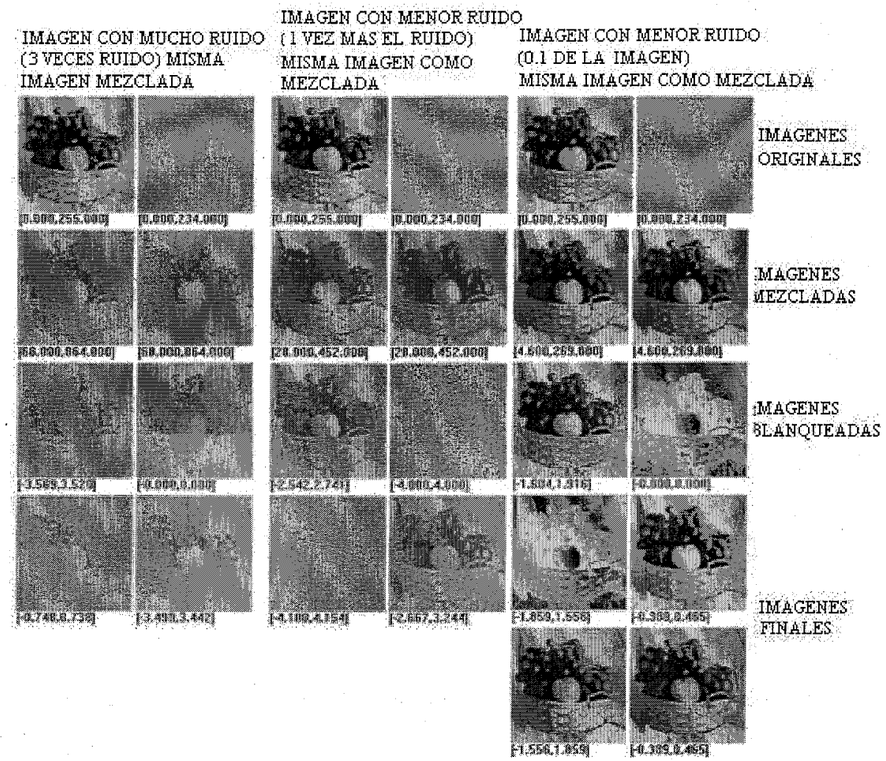


Figure 5-14: Mezcla de una imagen con diferentes niveles de ruido.

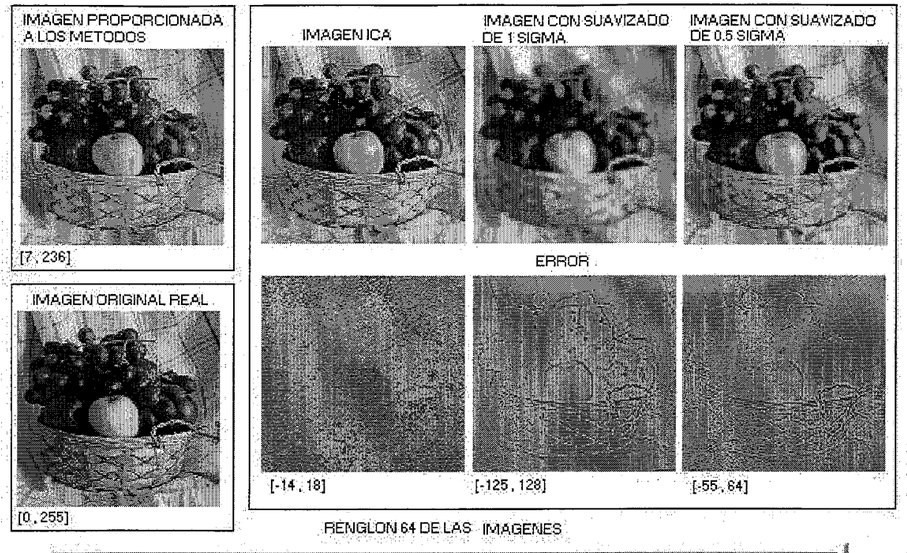


Figure 5-15:

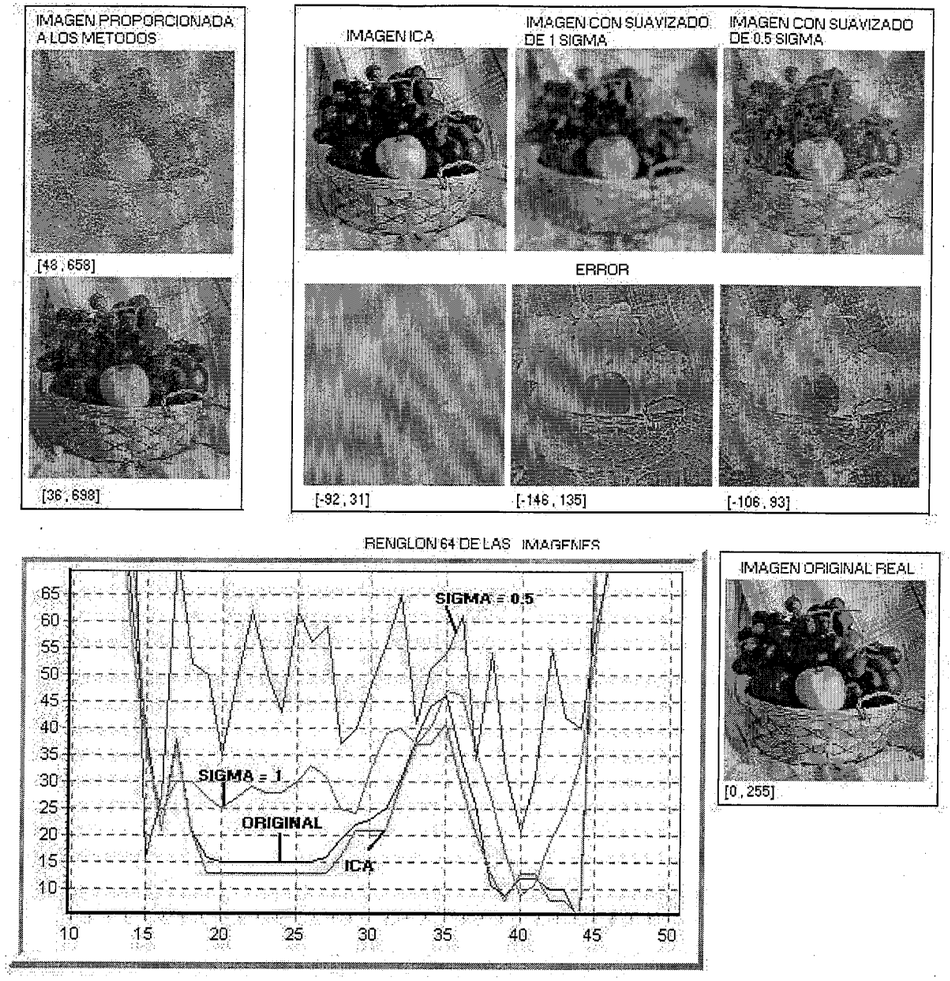


Figure 5-16:

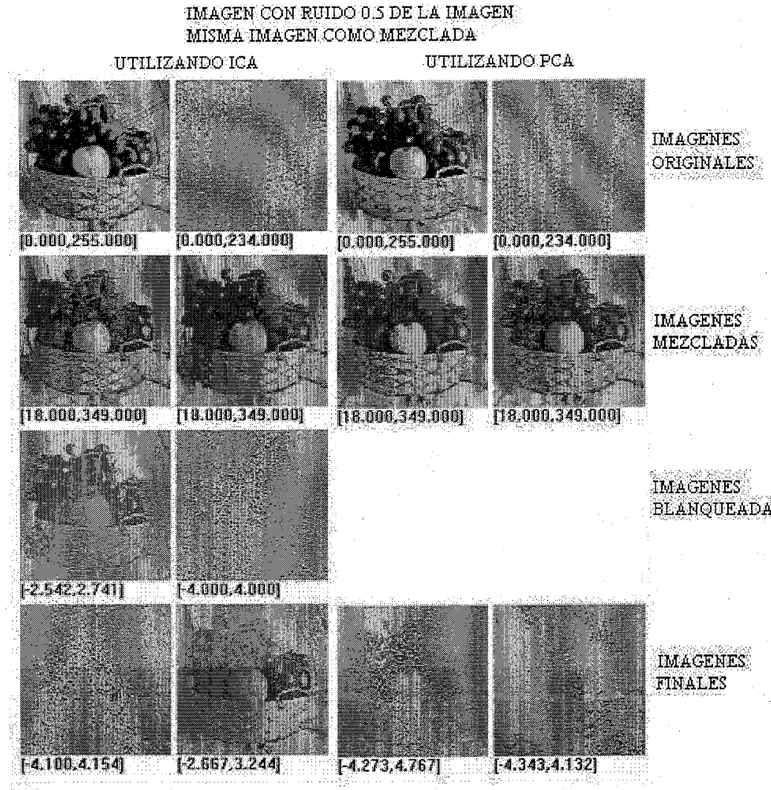


Figure 5-17: Comparación entre PCA e ICA

Chapter 6

ICA PARA CLASIFICACIÓN POR REDUCCION DE LA DIMENSIÓN(PROPUESTA)

Un objetivo importante en el análisis de datos multivariados es la clasificación de éstos en categorias con la meta de hacerlo con el menor error posible y en poco tiempo. Mientras más grande sea la dimensión, mayor será el tiempo necesario para la clasificación, por lo que muy ligado con este problema es la reducción de la dimensión. Suponiendo que tenemos un conjunto que consiste en un gran número de datos, denotemos el número de variables por m, el número de observaciones por n. ¿Cómo podemos transformar un conjunto de datos de dimensión m a dimension r, donde $r < m$, sin que se pierda información esencial para la clasificación? Esto se puede lograr si encontramos "factores" o "proyecciones" que nos den la mayor cantidad de información posible. De ésta manera, el objetivo de esta sección es clasificar un conjunto de datos a través de una proyección en un espacio de una dimensión menor a la original para así reducir el tiempo de cómputo. Para seleccionar las proyecciones se buscarán las menos gausseanas tomándolas como las mas "interesantes", de forma semejante a lo que se hace en "projection pursuit", ésto es, los componentes independientes.

Una vez que se tienen las observaciones en una dimensión menor, se aplica el método de vecinos más cercanos para la clasificación, ya es uno de los mejores para los conjuntos de datos

utilizados [5]; además de su sencillez de programación. Todo éste capítulo propone opciones de utilización de ICA para la reducción de la dimensión y para la clasificación.

6.1 CONJUNTOS DE DATOS

En ésta sección se describen las características de los conjuntos de datos utilizados para la clasificación. Todos ellos publicados por diferentes grupos de investigación.

6.1.1 DATOS DE SEGMENTACIÓN DE IMAGENES (SEGMENT.DAT)

El creador de este conjunto de datos fue el Grupo de Visión de la Universidad de Massachusetts (*Carla Brodley, brodley@cs.umass.edu).*

El archivo consta de 2310 observaciones obtenidas aleatoriamente de una base de datos de 7 imágenes previamente segmentadas a mano por pixel. La distribución de las clases es de la siguiente manera: (1) construcciones de ladrillos, (2) cielo, (3) follaje, (4) cemento, (5) ventanas, (6) caminos y (7) hierba. Cada observación representa una región de 3x3 pixeles por medio de 19 atributos continuos (descritos en el Apéndice E).

Para realizar las pruebas de clasificación, se dividió el conjunto original de dos diferentes maneras distribuidas como sigue:

CLASE	1	2	3	4	5	6	7	Total
Entrenamiento 1	281	277	286	272	284	281	285	1966
Prueba 1	49	53	44	58	46	49	45	344
Entrenamiento 2	229	231	233	219	231	227	235	1605
Prueba 2	101	99	97	111	99	103	95	705

La distribución de los datos para las primeras 7 variables se presenta en la Fig. 6.1.

6.1.2 DATOS DE LAS PRESENCIA DE DIABETES EN LOS INDIOS DE LA TRIBU PIMA (DIABETES.DAT)

El creador de la base de datos es el Instituto Nacional para las Enfermedades de Diabetes y Digestión de Kidney; el donador Vincent Sigillito (*vgs@aplcen.apl.jhu.edu*). Los pacientes de esta base fueron mujeres de al menos 21 años de edad descendientes de los indios Pima. El número de observaciones de la base de datos es de 768 con 8 atributos diferentes por renglón y

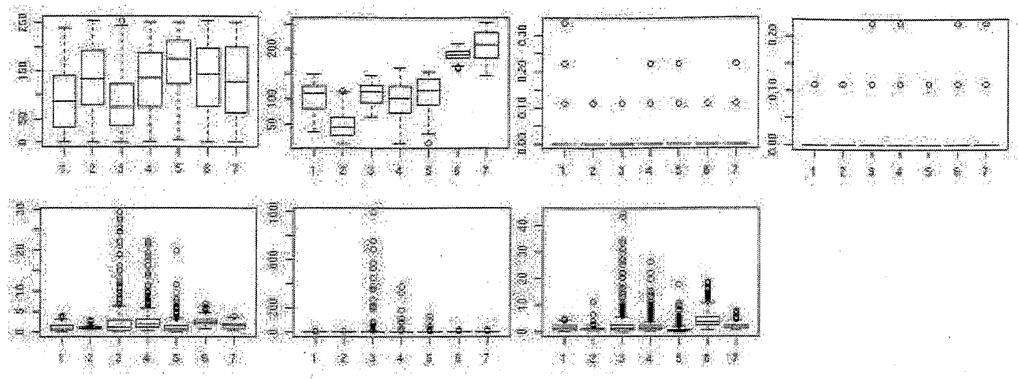


Figure 6-1: Distribución de los datos de acuerdo a las primeras 7 variables.

la clase a la que pertenece, donde el identificador 1 representa que la prueba de la diabetes es positiva y 0 negativa. Los atributos que se contemplan se encuentran descritos en el Apéndice E. Las distribuciones de las primeras cuatro variables se muestran el la figura 6.2.

Las observaciones se encuentran divididas en dos archivos de la siguiente manera:

	Clase		
Archivos	0	1	Total
Entrenamiento 1	334	178	512
Prueba 1	166	90	256
Entrenamiento 2	374	202	576
Prueba 2	126	66	192

en el archivo 2, en aquellas columnas de las cuales no se conocia el valor (primeramente tomadas como 0), se sustituyeron por la media de la columna.

6.1.3 CONJUNTO DIGITOS MANUSCRITOS (ZIP.DAT)

El conjunto de datos se compone de dígitos manuscritos normalizados, escaneados de los sobres del Servicio Postal de los U.S.A. Los dígitos originales son binarios y de diferentes tamaños y orientaciones; las imágenes están normalizadas a 16 x 16 pixeles en escalas de grises (Fig.6.3).

Los datos se encuentran en dos archivos; cada línea consiste en el identificador del digito

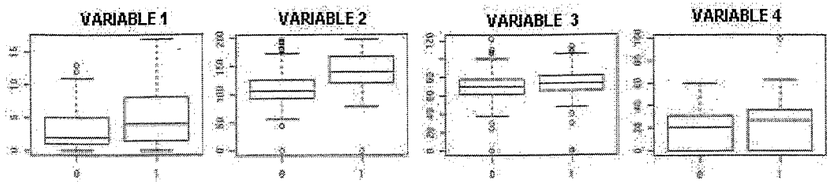


Figure 6-2: Primeras cuatro variables

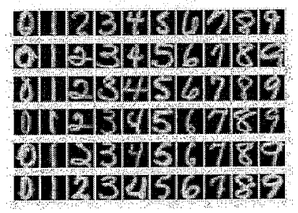


Figure 6-3: Ejemplos de digitos de la base de datos zip.dat

(0-9) seguido por 256 valores que representan la escala de gris de cada pixel de la imágen.

Las observaciones en los archivos se encuentran distribuidas como sigue:

Archivos	0	1	2	3	4	5	6	7	8	9	Total
Entrenamiento	1194	1005	731	658	652	556	664	645	542	644	7291
Prueba	359	264	198	166	200	160	170	147	166	177	2007
Prueba Nueva	332	249	190	157	212	174	199	163	151	173	2000

o como proporciones

ARCHIVOS	0	1	2	3	4	5	6	7	8	9
Entrenamiento	0.16	0.14	0.1	0.09	0.09	0.08	0.09	0.09	0.07	0.09
Prueba	0.18	0.13	0.1	0.08	0.10	0.08	0.08	0.07	0.08	0.09
PruebaNueva	0.17	0.12	0.09	0.08	0.11	0.09	0.10	0.08	0.07	0.08

Como información adicional, se sabe que el conjunto de prueba es notoriamente dificil, y un promedio de error de 2.5% es excelente [1]. Las distribuciones de los datos de las variable 5-8 se presentan en la Fig.6.4.

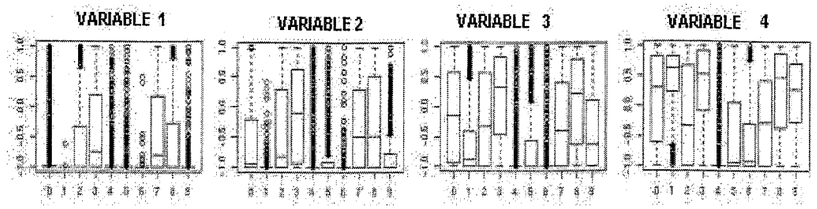


Figure 6-4: Distribución de los datos de las variables 5-8 de los datos

6.2 CLASIFICACIÓN DE DATOS POR VECINOS MÁS CERCANOS

Existen dos tipos de algoritmos de clasificación: los algoritmos supervisados y los no supervisados (Apéndice D). En la clasificación supervisada se tiene suficiente información de la clase a la que pertenecen algunos datos (entrenamiento). Una vez que se ha entrenado al clasificador, se prueba con un conjunto diferente para ver su efectividad (prueba). Los no supervisados no requieren de información a priori de las clases, como el clustering. Como ya se mencionó, en investigaciones anteriores [5] se puede verificar que el método de vecinos más cercanos es el que mejor clasifica a los conjuntos utilizados, además de su sencillez de programación, por lo que fue el que se implementó. Una de sus desventajas principales es el tiempo requerido de cómputo.

6.2.1 RESULTADOS

ARCHIVO	% CLASIFICACION
Segment.dat	95.35%
Zip I	94.37%
Zip II	96.45%
Diabetes I	69.14%
Diabetes II	72.39%

Un ejemplo del comportamiento de como se realiza la clasificación se muestra en la figura 6.5.

A continuación se presentan diferentes propuestas de ICA aplicado a clasificación.

DATOS SIN PROYECTAR											
	Clase 0	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5	Clase 6	Clase 7	Clase 8	Clase 9	
Clase 0	355	0	2	0	0	0	0	1	0	1	359
Clase 1	0	255	0	0	6	0	2	1	0	0	264
Clase 2	6	1	183	2	1	0	0	2	3	0	198
Clase 3	3	0	2	154	0	5	0	0	0	2	166
Clase 4	0	3	1	0	182	1	2	2	1	8	200
Clase 5	2	1	2	4	0	145	2	0	3	1	160
Clase 6	0	0	1	0	2	3	154	0	0	0	170
Clase 7	0	1	1	1	4	0	0	139	0	1	147
Clase 8	5	0	1	5	1	1	0	1	148	3	166
Clase 9	0	0	1	0	2	0	0	4	1	169	177
	371	261	194	167	198	155	160	150	156	185	2007
Clase 0	0.9888	0	0.00557	0	0	0	0	0.00279	0	0.00279	
Clase 1	0	0.96591	0	0	0.02273	0	0.00758	0.00379	0	0	
Clase 2	0.0303	0.00505	0.92424	0.0101	0.00505	0	0	0.0101	0.01515	0	
Clase 3	0.0181	0	0.01205	0.9277	0	0.03012	0	0	0	0.01205	
Clase 4	0	0.015	0.005	0	0.91	0.005	0.01	0.01	0.005	0.04	
Clase 5	0.0125	0.00625	0.0125	0.025	0	0.90625	0.0125	0	0.01875	0.00625	
Clase 6	0	0	0.00588	0	0.01176	0.01765	0.90471	0	0	0	
Clase 7	0	0.0068	0.0068	0.0068	0.02721	0	0	0.94558	0	0.0068	
Clase 8	0.0301	0	0.00602	0.0361	0.00602	0.00602	0	0.00602	0.89157	0.01807	
Clase 9	0	0	0.00565	0	0.0113	0	0	0.0226	0.00565	0.9548	
	0.1849	0.13004	0.09666	0.0832	0.09865	0.07723	0.0847	0.07474	0.07773	0.09218	94.37%

Figure 6-5:

6.3 PRIMERA PROPUESTA A CLASIFICACION (NN1A)

Desde hace varias décadas, varios investigadores (mencionados en [3]), han desarrollado técnicas de reducción de dimensión; PCA es una de ellas. Así, n variables aleatorias correlacionadas son tranformadas en un conjunto de $r \leq n$ variables no correlacionadas que son combinaciones lineales de las originales y pueden ser utilizadas para expresar los datos en una forma reducida. La manera de reducir la dimensión es obteniendo los componentes principales, ordenándolos de mayor a menor conforme los valores propios, y posteriormente proyectar los datos sobre los primeros vectores. Los eigenvectores se ordenan debido a que el error que se comete al eliminar una proyección sobre un componente principal es proporcional al valor propio correspondiente; tomándo así las direcciones con mayor variabilidad.

De manera similar a lo anterior se busca reducir la dimensión de los datos pero con ICA. Es decir, obtener un conjunto formado por r componentes independientes, proyectar sobre ellos los datos, efectuar la clasificación con vecinos más cercanos y verificar el porcentaje de clasificación correcta. Como se mencionó en el capitulo I, se tiene la ambiguedad de que una vez que se han obtenido los componentes independientes, no se puede determinar el orden de ellos[Capítulo 2].

Por lo que una propuesta, es evaluar cada vector que representa un componente independiente, obtener los valores de kurtosis y de negentropía asociados y en base a esta medida ordenarlos; es decir, haciéndolo de acuerdo a la dirección de máxima no gausseanidad.

6.3.1 PRUEBAS SEGMENT

En éstas pruebas se dejaron correr los algoritmos hasta tener 10 minutos de no hallar algún componente, ya que se comprobó que en la mayor parte de los casos, después de 5 min.de espera no se logran mejores resultados en cuanto al número de componentes encontrados.

En ésta sección se escogieron arbitrariamente 50 archivos de entrenamiento basados en el conjunto original y tomando 1900, 1700 y 1500 muestras. Una vez que se encontraron los componentes independientes, se proyectaron los datos sobre ellos y se calculó el promedio de error, obteniéndose:

1900 datos	3	4	5
PCA	72.60%(3.42%)	91.41%(1.69%)	91.62%(1.70%)
P.F.Kurtosis	83.03% (5.97%)	88.67%(3.58%)	90.99%(2.84%)
P.F.Negentropía	80.95%(7.15%)	85.87%(4.26%)	87.85%(3.88%)

1700 datos	3	4	5
PCA	72.14%(4.93%)	91.14%(1.66%)	91.37%(1.57%)
P.F.Kurtosis	83.05% (4.48%)	87.63%(4.50%)	90.29%(3.60%)
P.F.Negentropía	80.02%(6.08%)	85.42%(5.21%)	87.77%(3.75%)

1500 datos	3	4	5
PCA	50.35%(33.06%)	58.24%(32.55%)	58.23%(32.55%)
P.F.Kurtosis	51.81%(30.84%)	55.57%(31.88%)	52.09%(24.94%)
P.F.Negentropía	45.38%(24.08%)	53.06%(30.97%)	53.52%(29.79%)

Lo anterior se hizo para conocer la incertidumbre del error, ya que solo de ésta manera se pueden comparar los desempeños de NN y de NN1A. La confiabilidad de los intervalos es del 90%. La variabilidad de las clasificaciones depende mucho de cuáles componentes se escogen y no de cuál de los componentes tiene mayor valor de no gausseanidad. A continuación se realizan las pruebas para los archivos propuestos en particular.

ARCHIVO 1

Utilizando los métodos de FastICA con medición de gausseanidad con kurtosis y con negentropía se obtuvieron únicamente 5 componentes independientes en ambos casos). Los porcentajes de clasificación se resumen en la tabla

	Componentes sobre los que se proyecta		
Método	3	4	5
PCA	71.80%	91.57%	91.86%
FastICA Kurtosis	76.16%	77.62%	91.57%
FastICA Negentropia	76.86%	85.46%	89.53%

Se puede observar que utilizando únicamente 5 componentes independientes y proyectando los datos sobre ellos se tiene una clasificación buena comparada con la clasificación con NN directamente, ésto es muy importante, ya que se se están utilizando vectores de dimensión mucho menor a la original por lo que se ésta reduciendo el tiempo de ejecución de NN. El desempeño de ICA, con ambos métodos, es menor a PCA para clasificar los datos, pero se incrementa conforme menos componentes son considerados. En la Fig.6.6 se observan las distribuciones de los datos usando las proyecciones sobre los c.i. calculadas de manera diferente.

ARCHIVO II

Al igual que en la prueba anterior, se obtienen únicamente 5 componentes independientes, mostrándonse algunos ejemplos en la tabla:

	Componentes sobre los que se proyecta		
Metodo	3	4	5
PCA	81.42%	86.52%	90.21%
FastICA Kurtosis	70.86%	71.42%	90.07%
FastICA Negentropia	76.17%	78.31%	82.70%

Aqui se puede ver que con 5 componentes los resultados son similares, pero con la diferencia

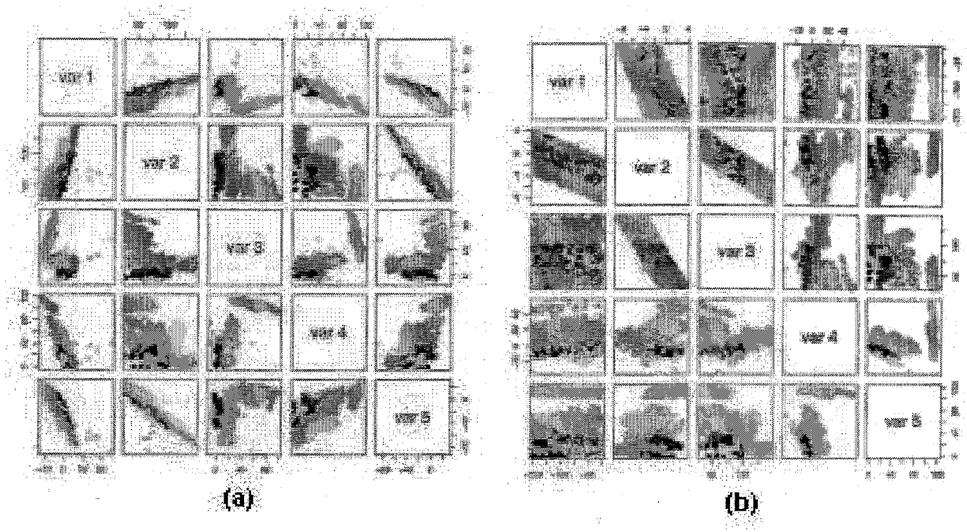


Figure 6-6: Proyecciones sobre 5 componentes independientes del archivo de entrenamiento I calculados con los métodos de (a) FastICA con Kurtosis (b)FastICA con Negentropia.

de que el CI no supera la clasificación con PCA. En los archivos anteriores puede verse que mientras menos datos se encuentren en el archivo de entrenamiento, la clasificación empeora considerablemente para ICA y PCA.

6.3.2 PRUEBAS DIABETES EN PIMA INDIANS

Al igual que en el conjunto de prueba anterior, el tiempo máximo de espera para encontrar un componente fue de 10 min. Los resultados obtenidos se muestran en la siguiente tabla:

	Metodo (ARCHIVO I)			Metodo(ARCHIVO II)		
C.I	PCA	FastICA Kurtosis	FastICA Negent.	PCA	FastICA Kurtosis	FastICA Negentrop.
7	69.14 %	70.70 %	69.53 %	73.44 %	70.83 %	72.40 %
5	70.70 %	74.22 %	73.05 %	69.27 %	68.75 %	66.15 %
4	68.75 %	71.87 %	66.40 %	67.18 %	67.19 %	67.18 %
3	65.62 %	64.40 %	70.71 %	62.50 %	72.91 %	67.70 %
2	63.67 %	64.02 %	65.62 %	66.14 %	53.10 %	66.14 %

Se puede ver una mejora en el porcentaje de clasificación, aumentandose de 67.58% (NN directamente) a porcentajes de incluso 74.22% (en el mejor de los casos) con dos variables menos para representar cada dato y el metodo basado en kurtosis y 73.05% con negentropía e incluso cabe señalar el 72.91% con únicamente 3 componentes.

6.3.3 PRUEBAS ZIP

Aplicando el algoritmo de FastICA para encontrar los componentes independientes en el conjunto de entrenamiento, se obtuvieron 28 con la medición de gausseanidad por kurtosis y 96 con negentropia (con un tiempo límite de aproximadamente 12 horas para encontrar los componentes).Graficando los primeros ocho componentes independientes sobre los que se realizó la proyección de los datos de prueba se obtiene la figura 6.7. La tabla de clasificación se presenta a continuación.

CLASIFICACION DE LOS DATOS ZIP_TEST UTILIZANDO PROY SOBRE 20 COMPONENTES INDEPENDIENTES											
	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5	Clase 6	Clase 7	Clase 8	Clase 9	Clase 10	
Clase 1	**156**	27	8	20	19	14	38	12	18	20	332
Clase 2	3	**231**	0	2	2	0	0	1	2	8	249
Clase 3	28	0	**113**	9	8	10	7	3	2	10	190
Clase 4	20	5	9	**71**	4	12	9	17	5	5	157
Clase 5	25	9	8	8	**87**	14	13	14	6	28	212
Clase 6	25	2	2	11	17	**78**	7	13	7	12	174
Clase 7	18	16	7	5	8	7	**105**	14	7	12	199
Clase 8	18	9	4	7	5	3	2	**96**	6	13	163
Clase 9	22	4	0	8	17	6	7	18	**60**	9	151
Clase 10	11	33	0	2	11	4	3	11	8	**90**	173
	326	336	151	143	178	148	191	199	121	207	**0.5435**

Figure 6-7:

% DE CLASIFICACION PARA ARCHIVO DE PRUEBA I							
Metodo		Numero de componentes para proyección					
		1	15	17	20	28	
PCA		35.77			93.77	94.77	
FastICA	Kurtosis		33.50	35.45	36.10		
	Kurt.Comp		76.98	79.92	81.86		

% DE CLASIFICACION PARA ARCHIVO DE PRUEBA II							
Metodo		Numero de componentes para proyección					
		1	15	17	20	28	96
PCA		54.89			95.42	96.11	94.47
FastICA	Kurtosis		53.05	54.45	56.85		
	Kurt.Comp		83.00	85.50	87.65		

Para este conjunto de datos el blanqueado no proporciona un algoritmo eficiente de clasificación, por lo que se utilizó el algoritmo de FastICA con kurtosis pero sin la suposición de decorrelación (ver deducción en Apéndice B) llamandole FastICA con Kurtosis Completo.

El número de componentes que se eligió como base fue de 17 (± 2 componentes), en general se tiene el 20% de mala clasificación, aumentando este número de componentes no se da una mejora considerable. Al realizar estas pruebas, en términos generales el dígito peor clasificado

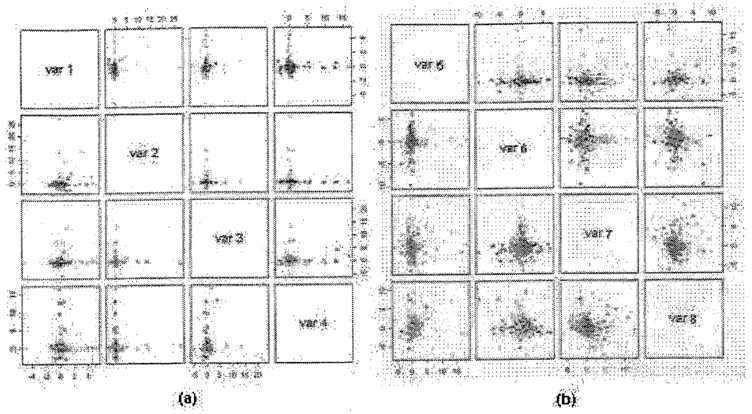


Figure 6-8: Componentes que se utilizaron para proyectar los datos zip_test con 20c.i. (a) 1-4 (b) 5-9.

es el "8 "; ya que se confunde con el "3 "(en 1.3% de las veces), con el "2" (1.15%) y con el "9" (1.05%). El "9" es el segundo dígito que se confunde (con el "7 " el 0.99% y con el "4" el 1.4%). Al contrario, el "1", es en general el mejor clasificado con un porcentaje superior al 94.00%.

Partiendo de las pruebas anteriores, se puede ver un mejor desempeño del método de FastICA con kurtosis, por lo cual las siguientes modificaciones únicamente se hicieron para éste método. También se observó en éstas pruebas que el desempeño del clasificador depende mucho de los componentes que se escojan, pero no puede concluirse que tomando aquéllos que presentan una mayor no gausseanidad nos proporcionen mejores resultados. También el punto de arranque influye en sobremanera en los resultados obtenidos, por lo que se probaron puntos de arranque aleatorios, fijos e iguales a un componente PCA; también se forzaron algunas entradas de vectores que eran muy pequeñas para que fueran cero, obteniéndose mejoras en algunos casos, pero no siempre.

6.4 SEGUNDA PROPUESTA DE CLASIFICACION (NN2A)

La segunda propuesta para mejorar los resultados es modificar el algoritmo de FastICA con medición de no gausseanidad con kurtosis, de tal manera que en lugar de buscar proyecciones que maximizan la no gausseanidad sobre TODOS los datos, se consideren también las no gausseanidades de los datos que pertenecen a una misma clase (la cual se quiere minimizar).

La búsqueda de los componentes independientes está ligada a la maximización del valor absoluto de la no gausseanidad (Capítulo 3)

$$\max(U = |Kurt(w^T x)|)$$

Modificando ésta función de optimización para que se minimicen los componentes de cada una de las clases que componen al conjunto original, se tiene:

$$U = \left|Kurt(w^T x)\right| - \sum_{i=1}^{k} \lambda_i \left|Kurt_i(w^T x)\right|$$

donde k es el número de clases y $Kurt_i(w^T x)$ es el valor de la kurtosis de la distribución de los datos $w^T x$ donde se restringe x a los datos que pertenecen a la clase i. Partiendo de la deducción mostrada en el Apéndice A, se tiene que el gradiente del valor absoluto de la kurtosis por medio de punto fijo de $w^T z$ se puede calcular como:

$$w \propto \left[E\left\{z\left(w^T z\right)^3\right\} - 3||w||^2 w\right]$$

introduciendo la modificación propuesta, se tiene:

$$w \propto \left[E\left\{z\left(w^T z\right)^3\right\} - 3||w||^2 w\right] - \sum_{i=1}^{k} \lambda_i \left[E_i\left\{z_i\left(w^T z_i\right)^3\right\} - 3||w||^2 w\right]$$

Ésta ecuación sugiere un algorimo de punto fijo de la siguiente manera:

$$w \leftarrow \left[E\left\{z\left(w^T z\right)^3\right\} - 3w\right] - \sum_{i=1}^{k} \lambda_i \left[E\left\{z_i\left(w^T z_i\right)^3\right\} - 3w\right]$$

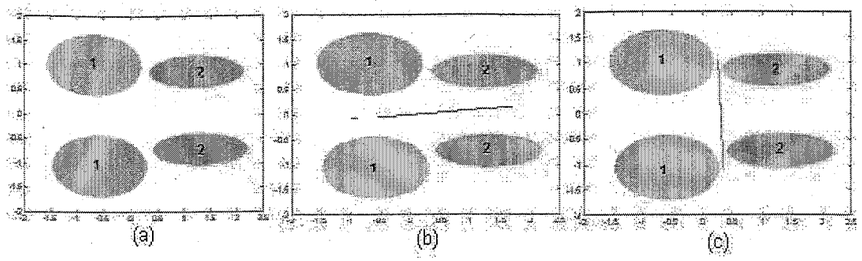


Figure 6-9: Mejora al algoritmo de FastICA con kurtosis.

$$w \leftarrow w/norm(w)$$

si las λ_i son iguales:

$$w \longleftarrow E\left(z\left(w'z\right)^3\right) - 3w - \lambda \sum_i \left[E\left(z_i\left(w^T z_i\right)^3\right) - 3w\right]$$

De esta manera se puede cambiar el orden en el que se encuentran los componentes independientes. Supongamos que tenemos la distribución de clases mostrada en la Fig. 6.9a. El primer componente que se encuentra por el método de FastICA no es él que clasifica mejor (Fig. 6.9b); aplicando la modificación propuesta al algoritmo, se encuentra el vector mostrado en la Fig. 6.9c; el cual sin duda, es mejor ya que se tienen separadas ambas clases utilizando únicamente el primer componente independiente.

6.4.1 PRUEBAS SEGMENT

ARCHIVO I

En esta prueba se utilizó la modificación anterior al método de FastICA con kurtosis.

	PARÁMETRO λ							
Num. de comp que se utilizaron	0.009	0.01	0.019	0.02	0.025	0.5	1.0	0.001
5	90.41	-	-	-	92.15	91.28	91.28	91.56
4	89.24	89.24	-	-	70.64	92.15	86.63	81.10
3	87.79	76.45	77.33	77.03	63.66	87.79	74.42	78.79
Max.num. de comp. que se encuentran	5	4	3	3	5	5	5	5

ARCHIVO II

	PARAMETRO λ							
Num. de comp. que se utilizaron	0.009	0.01	0.019	0.02	0.025	0.5	1.0	10.0
5	90.07	90.07	90.21	90.21	90.36	89.93	89.79	89.64
4	60.00	60.43	59.72	59.57	64.54	87.21	88.65	88.65
3	57.30	56.60	51.91	53.19	62.82	83.83	86.10	81.84
Max.num.de comp. que se encuentran	5	5	5	5	5	5	5	5

Como puede observarse con 5 componentes se logra una mejora de 1% aproximadamente, pero lo relevante es que también con 4 componentes se logra dicho porcentaje de clasificación, lo que mejora los resultados obtenidos anteriormente incluso mejora a PCA que para este tipo de datos es el que da mejor clasificación.

6.4.2 CONJUNTO DIABETES

Los resultados se encuentran resumidos en la siguiente tabla:

ARCHIVO I

	λ										
Num Comp.	0.01	0.016	0.02	0.03	0.04	0.07	0.15	0.2	0.25	0.3	2.0
6	69.14	68.75	68.75	67.18	68.35	63.67	69.14	**69.92**	68.75	67.97	**69.92**
5	67.96	72.26	66.79	68.35	66.01	66.79	**72.66**	71.09	67.57	68.35	67.18
4	62.11	64.45	62.89	64.84	64.84	63.67	64.84	62.50	62.50	**66.01**	58.59
3	60.93	64.84	59.76	59.37	57.42	58.98	59.37	60.16	63.28	63.67	**65.23**
2	**64.06**	58.98	63.28	**64.06**	59.76	59.37	59.37	57.81	57.42	57.42	60.93

ARCHIVO II

Comp.	0.01	0.016	0.02	0.03	0.04	0.07	0.1	0.125	0.2	0.25	0.3	2.0
6	**73.96**	-	-	69.79	-	67.19	69.27	70.31	65.62	72.39	70.83	70.83
5	68.23	-	-	67.71	**73.95**	65.62	70.83	72.39	68.23	71.35	65.62	71.87
4	71.88	-	-	69.27	70.83	70.83	72.91	**77.60**	65.10	68.23	66.14	68.29
3	**71.87**	63.54	68.23	66.67	66.14	69.27	68.15	71.35	62.50	**71.87**	64.06	63.54
2	64.58	60.90	71.35	61.46	68.23	70.83	**72.92**	69.27	62.50	64.07	65.10	59.37

En estas pruebas se observa una mejora de 1% cuando se utilizan 6 componentes, con 5 se conservan más o menos los mismos porcentajes. Pero cabe señalar que con 4 logra un porcentaje de 77.60%. lo cual es bastante alta, incluso superior a NN y disminuyendose el tiempo de clasificación en un 50%

6.4.3 CONJUNTO ZIP

Realizando pruebas asignando valores diferentes de lambda; se considera que el número de componentes que se van a utilizar son 17. Cada prueba se dejó correr un tiempo fijo entre 45 min. y 1 hora, y no se obtuvieron buenos resultados, ya que despues de variar diferente numero de parámetros no se consiguió ninguna mejora.

Como puede observarse en todas las pruebas indicadas, se pueden lograr mejoras en los porcentajes siempre y cuando se encuentren los parámetros adecuados.

6.5 TERCERA PROPUESTA DE CLASIFICACION (NN3A)

En el caso del método de FastICA con kurtosis, se desea que el valor absoluto de la kurtosis se incremente. La propuesta es hacer que el parámetro λ anterior vaya aumentando gradualmente un ε conforme pasa el tiempo de manera similar a la estrategia que utiliza el método de recocido simulado (simulated annealing). La actualización del parámetro se hace para cada paso del algoritmo FastICA.

6.5.1 CONJUNTO SEGMENT

Utilizando diferentes valores de λ y de ε se obtiene la siguiente tabla:

% CLASIFICACION (ARCHIVO I)				
	λ			
ϵ	0.2	0.3	0.5	1.0
0.05	71.63(3) 75.85(4) 90.07(5)	81.56(3) 89.50(4) 91.63(5) 93.19(6) 95.17(7)	89.95(3) 89.50(4) 91.63(5) 94.33(6) 95.18(7)	69.21(2)
0.01	59.85(3) 61.42(4) 90.07(5)	59.85(3) 61.42(4) 59.85(5)	59.85(3) 61.42(4) 90.07(5)	59.85(3) 61.42(4) 90.07(5)
0.07(2)	-	-	-	-
0.55(2)	-	-	-	-

% CLASIFICACION (ARCHIVO II)				
	λ			
ϵ	0.009	0.01	0.5	1.0
0.05	64.58(3) 81.98(4) 95.34(5)	80.81(3) 81.10(4) 91.27(5) 93.84(6)	88.36(4) 92.76(5) 93.04(6)	80.56(4) 85.39(5)
0.01	70.60(3) 72.05(4) 83.40(5) 88.36(6) 93.90(7)	53.19(3) 75.03(4) 77.16(5)	87.23(4) 93.61(5)	77.30(3) 80.14(4)
0.07	54.47(3) 62.27(4) 89.50(5)	51.48(3) 74.32(4) 78.15(5)	79.29(3) 86.38(4) 92.90(5)	63.26(3) 78.29(4)
0.55	51.18(3) 74.18(4) 78.43(5)	-	79.57(3) 84.96(4) 93.47(5)	75.31(3) 77.44(4) 91.63(5) 94.04(6)

Como se puede observar algunos de los porcentajes de clasificación obtenidos son superiores a los encontrados en las pruebas anteriores, pero el problema en esta propuesta es que se tienen que encontrar los valores de los parámetros, lo cual no siempre es sencillo.

6.5.2 CONJUNTO DIABETES

Los resultados se tienen en la siguiente tabla.

Parametros		Comp. p/proy		
λ	ϵ	6	5	4
0.1	0.001	60.94	61.32	60.94
	0.01	67.57	70.70	64.45
	0.05	63.28	64.45	62.50
	0.1	67.19	70.70	62.89
0.15	0.001	67.96	67.97	67.97
	0.01	65.23	64.45	64.06
	0.05	67.58	68.35	59.37
	0.1	69.53	70.71	60.93

Parametros		Comp. p/proy		
λ	ϵ	6	5	4
0.2	0.01	62.10	62.50	62.50
	0.05	65.62	69.92	66.40
	0.1	71.88	72.27	63.28
	0.5	66.79	66.79	66.79
1.0	0.001	62.50	61.32	63.28
	0.01	58.21	60.93	62.89
	0.05	61.72	62.11	60.16
	0.1	61.32	62.89	60.54

Para el archivo II

		Comp. p/proyectar					
λ	ϵ	7	6	5	4	3	2
0.1	0.001	-	-	-	66.67	66.67	75.00
	0.01	71.35	70.31	71.87	65.62	65.10	62.50
	0.05	-	-	-	-	67.71	70.31
0.15	0.001	72.92	71.88	72.92	77.60	68.75	65.10
	0.01	-	-	-	72.39	64.06	64.58
	0.05	-	-	-	-	70.31	61.97
	0.1	-	73.96	70.83	70.31	65.62	60.42
	0.5	-	70.83	70.31	70.83	66.15	67.71
0.2	0.001	72.92	72.39	72.91	70.31	67.19	67.19
	0.01	70.31	69.79	67.19	70.31	71.35	69.79
1.0	0.001	70.83	72.92	73.95	71.35	69.79	61.97
	0.01	72.39	67.71	68.23	67.19	61.98	60.42
	0.05	71.35	71.87	72.91	72.91	71.87	68.75

Como puede observarse no se presenta ninguna mejora considerable en los porcentajes de clasificación y la prueba anterior.

6.5.3 CONJUNTO ZIP

Al igual que con la prueba anterior no se logró ninguna mejora.

6.6 MODELO BAYESIANO PARA ICA

Suponiendo el problema típico de clasificación, tenemos n número de observaciones que se agrupan en t clases diferentes $(c_1, c_2, ...c_t)$ y dada una cierta observación x_i queremos encontrar la clase c_j a la que pertenece. Una de las maneras mas conocidas para resolver el problema anterior es por medio de un clasificador bayesiano, por lo que en ésta sección se muestra uno basado en componentes independientes.

El teorema de Bayes dice:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

en nuestro caso en especial se tiene que

$$P(c_j|x_i) = \frac{P(x_i|c_j)P(c_j)}{P(x_i)}$$

tomando el denominador como un parámetro de regularización:

$$P(c_j|x_i) = \frac{P(x_i|c_j)P(c_j)}{Z}$$

donde la probabilidad de cada clase $P(c_j)$ se tiene de manera explícita, suponiendo que cada clase puede estar definida en término de sus componentes independientes, por lo que el estimador de la clase se calcula como:

$$c^* = \max_{c_j} [P(x_i|c_j)P(c_j)] \approx \max_{c_j} [F(x_i, c_j)P(c_j)]$$

donde $F(x_i, c_j)$ está definida en base a las proyecciones de los datos x sobre los componentes independientes de la clase c_j. Debido a que la distribución es discreta, y se necesita continua, se tomó una áproximación basada en kerneles. La idea general es la siguiente, en cada punto se

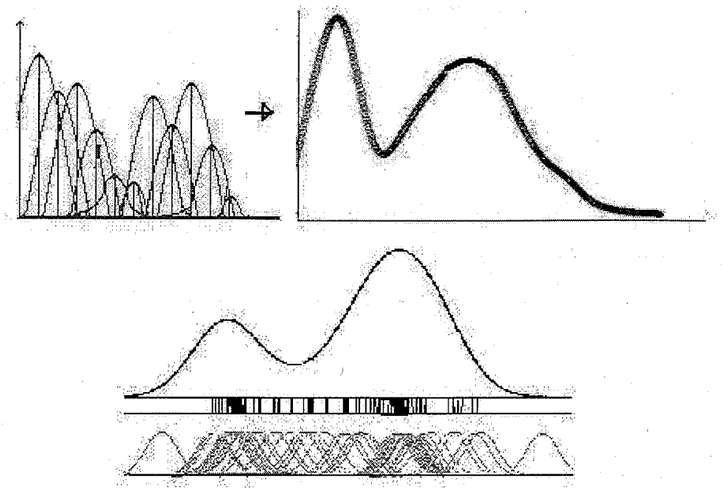


Figure 6-10: Aproximación basada en kerneles para una dimensión.

sobrepone un kernel gausseano, y posteriormente se hace el promedio entre ellos, obteniéndose curvas continuas que representan a la densidad de probabilidad.(Fig.6.10)

Entonces el algoritmo general es el siguiente:

1. Obtener los componentes independientes para cada clase c_j.
2. Realizar la proyección de todos los datos x del archivo de entrenamiento sobre los componentes independientes por clase.
3. Aproximar la verosimilitud por medio de kerneles gausseanos.
4. Para cada muestra del archivo de prueba, encontrar la posición con mayor probabilidad dentro de las densidades que representan a cada clase.
5. Calcular el porcentaje de clasificación.

Realizando el clasificador de Bayes se obtienen los siguientes resultados para cada archivo de prueba

SEGMENT (Archivo I)

Clasificador de Bayes tal cual= 85.46%

	10	7	5
ICA	85.47%(±2.45)	79.07%(±1.2)	77.32%(±3.67)
PCA	86.35%	80.23%	79.06%

SEGMENT (Archivo II)

Clasificador de Bayes tal cual= 87.23%

	10	7	5
ICA	82.13%(±3.02)	70.21%(±2.6)	66.24%(±2.56)
PCA	86.66%	80.56%	79.20%

ZIP

Clasificador de Bayes tal cual= 64.72%

	100
ICA	82.16%
PCA	91.33%

DIABETES (Archivo I)

Clasificador de Bayes tal cual= 76.56%

	10	7	5
ICA	67.96%(±5.02)	68.67%(±5.2)	59.38%(±2.05)
PCA	72.27%	67.97%	68.75%

DIABETES (Archivo II)

Clasificador de Bayes tal cual=72.40%

	10	7	5
ICA	67.39%(±2.3)	67.75%(±6.01)	63.40%(±8.45)
PCA	70.83%	74.47%	72.91%

Como puede observarse, no se pueden mejorar los porcentajes dados proyectando los datos sobre un numero de componentes menor al original, exceptuando en caso del conjunto de datos zip, en el que la mejora es impresionante; mucho mayor a cualquier clasificación realizada anteriormente, aunque el numero de componentes utilizados es mayor. En el caso de las pruebas anteriores se utilizaron kerneles gausseanos con parámetros fijos, aunque también se podrían variar.

En este capítulo se mostraron algunas propuestas de como utilizar ICA para clasificación, obteniéndose mejoras con cada una de las propuestas, aunque no siempre funcionan para

cualquier tipo de archivo. En forma particular, se encontró que en el archivo Zip no funcionaron los experimentos debido a que cada clase de datos se encuentran distribuidos de una manera muy cercana a la gausseana, lo cual no ocurre con los demás conjuntos de datos.

Chapter 7

APLICACION DE ICA EN EL MANEJO DE OTRO TIPO DE DATOS

En este capítulo se presenta una propuesta de aplicación de ICA para tiempos de respuesta entre computadoras por medio de pings. El objetivo de esta sección es mostrar un tipo de aplicación no muy explorada, que es el buscar una relación entre las máquinas que influyen en un ping y la red en la que se encuentran; en forma especial saber si la relación entre ellas puede manejarse por medio de componentes independientes.

Para esta aplicación se utilizaron los tiempos de respuesta entre un servidor de Unix y dos computadoras diferentes; realizando el ping desde el servidor durante 5 días consecutivos cada 5 segundos. Algunas de las series de tiempo se presentan en la Fig.7.1.

Se calcularon los componentes independientes por día, tomando en cada observación los tiempos de respuesta para cada una de las computadoras. Al igual que en los capítulos anteriores, se utilizó el algoritmo básico de ICA, mostrándose así su funcionalidad para series de tiempo. Debido a que las señales no están exactamente sincronizadas se tomaron las originales en las siguientes maneras para calcular los componentes:

1. Considerando toda la serie.

2. Sobre las diferencias del tiempo de respuesta actual sobre el anterior.

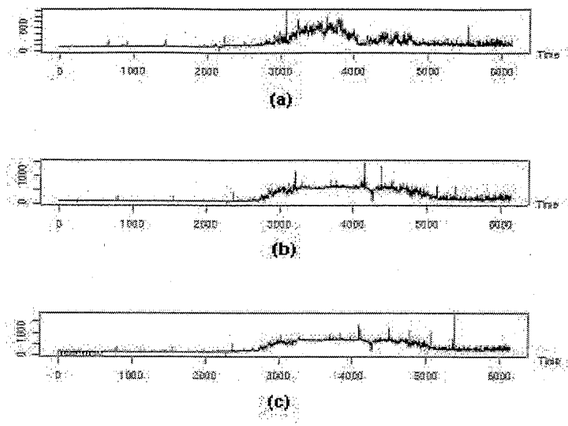


Figure 7-1: Señales originales de los tiempos de respuesta de las computadoras a los pings enviados desde un servidor Unix. (a) primera computadora, (b) segunda computadora y (c) tercera computadora.

3. Suavizando las señales.

4. Muestreando cada 5 o 3 datos.

5. Tomando la hora pico.

Finalmente, con la última propuesta los resultados que se obtuvieron mostraron que uno de los componentes independientes obtenidos es aproximadamente una combinación lineal de las señales originales (Fig. 7.2). Aunque en algunos casos para diferentes días se presentan componentes semejantes (Fig. 7.3); es decir, se puede ver una relación entre ellos, en la mayoría no se ve ningún componente igual a otro, probablemente porque no se trata de días con comportamiento "semejante". En aquellos casos en los que se tienen componentes similares, podria utilizarse los demás como indicadores de la red en la que se encuentran, aunque en este caso ésto no es muy claro por el poco número de datos con los que se cuenta.

Una posible manera de utilizar los componentes independientes para que presenten similitud en diferentes días es obteniéndolos para los tiempos de respuestas entre dos computadoras, y una vez que se tiene una muestra suficientemente grande, comparar los resultados para así

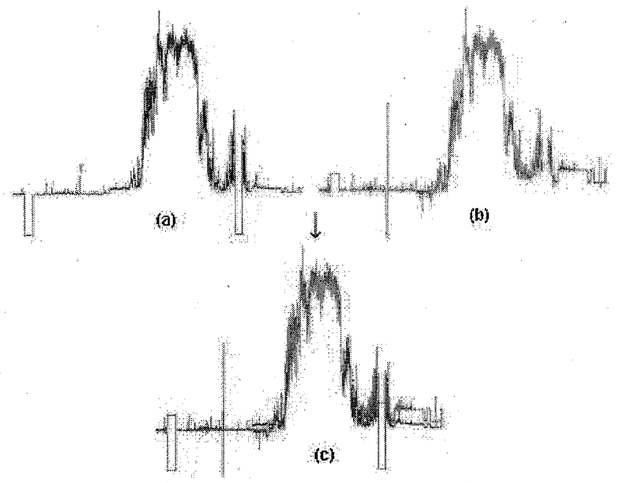


Figure 7-2: Señales de (a) combinación lineal de las señales originales, (b) primer componentes independiente, (c) comparación entre ellas.

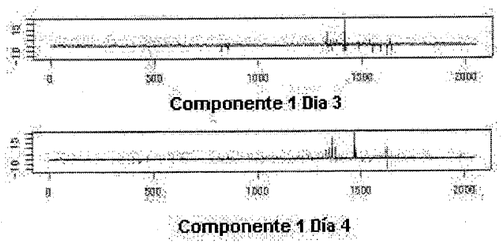


Figure 7-3: Primer componente enconntrado en diferentes días.

identificar cuando se presentó algun error o falla de medición. En el caso de esta tesis, ésta propuesta esta fuera de alcance y se presenta como una propuesta de trabajo futuro.

Chapter 8

CONCLUSIONES Y TRABAJO FUTURO

8.1 CONCLUSIONES

El estudio de ICA todavía es una rama no muy explorada, la cual ofrece grandes expectativas de crecimiento y aplicación. En términos generales en ésta tesis se siguieron dos caminos:

1. En la primera ruta se explicó la teoría de componentes independientes, haciéndo recopilaciones de diferentes fuentes; además de que se implementaron varios algoritmos para comprobar sus características de desempeño. Presentándose un documento completo y sencillo de entender para personas que no se encuentran familiarizadas con éste tema.

2. También se implementaron algunas variantes del método de ICA libre de ruido para su aplicación en procesamiento de imágenes, clasificación y series de tiempo.

Específicamente, se hicieron las siguientes aportaciones en la tesis:

1. Se presentó una propuesta de cómo utilizar ICA para la eliminación de ruido en imágenes de manera rápida y sencilla; con la característica principal de conservación de bordes y sin necesidad de modificar el algoritmo básico de FastICA (Capítulo 5).

2. Se comprobaron las características de los métodos, comprobando que los algoritmos se comportan de manera diferente dependiendo del punto de arranque; además de la velocidad de los algoritmos de punto fijo comparados con los basados en gradiente (Capítulo 3).

3. También se mostró a través de imágenes la aplicabilidad de ICA para separar diferentes señales que se encuentran mezcladas, sin necesidad de modificar el algoritmo básico (Capítulo 5).

4. En clasificación, se mostró por medio de ejemplos y contraejemplos que ICA es una posible alternativa en ésta área y en muchos casos superior a PCA siempre y cuando se tenga un gran número de datos con distribuciones no gausseanas, y se desee trabajar en un espacio de baja dimensión. Se propusieron cuatro vertientes:

   - La primera consistió en utilizarla para reducir la dimensión de varios conjuntos de datos, lo cual implica una reducción considerable en el tiempo de ejecución del método de clasificación, que en éste caso fue el algoritmo de vecino más cercano (propuesta NN1A).
   - La segunda fue realizando una extensión a un algoritmo conocido (algoritmo de punto fijo con medición de gausseanidad por kurtosis) para buscar además de los componentes que maximizan la no gausseanidad de todos los datos aquellos que pertenecen a una clase en específico (propuesta NN2A).
   - Como una mejora al método NN2A se presentó una idea basada en la estrategia del recocido simulado (propuesta NN3A).
   - Finalmente, se presentó un clasificador basado en el teorema de bayes, el cual provee resultados comparables a cualquier método de clasificación conocido; especialmente en los casos de que se tienen datos distribuidos gausseanamente, aún con la restricción básica de no manejar este tipo de distribuciones (Capítulo 6).

5. Como aplicación a series de tiempo se mostraron resultados generales para datos de redes, mostrándo que no es necesario modificar el algoritmo general para su utilización (Capítulo 7).

6. Finalmente, se proporcionó una manera de medir el desempeño del algoritmo en base al ángulo de los componentes en lugar de la matriz de permutación (Capítulo 3).

8.2 TRABAJO FUTURO

A continuación se desglosan brevemente algunos aspectos que se pueden ampliar de ésta tesis, destacando las áreas principales de desarrollo:

1. Para calcular los componentes principales (paso necesario antes de calcular los componentes independientes) se tridiagonalizó la matriz y se obtuvieron los eigenvalores y eigenvectores. Así que desde el punto de métodos numéricos, podría buscarse un algoritmo más eficiente para lograr una mayor velocidad de cálculo.

2. También en dicha área, podrían implementarse algunas modificaciones para mejorar los métodos de gradiente (en velocidad y convergencia).

3. Para mejorar el estudio de componentes independientes se tendrían que implementar todos los algoritmos que calculan componentes independientes para comparar sus desempeños.

4. Como se probó, con algunos datos, las clasificaciones funcionan mejor para datos no decorrelacionados, por lo que se podrían modificar los algoritmos de punto fijo y gradiente para no utilizar este prerequisito, cómo se hizo en el método de punto fijo basado en kurtosis.

5. En la sección de clasificación, se podrían utilizar los algoritmos de gradiente ya modificados para poder introducir conocimiento del comportamiento de los datos.

6. Si se tuviera como objetivo únicamente encontrar un algorimo de clasificación eficiente, se podrían implementar otros métodos y no solamente el de vecino mas cercano, ofreciéndose una mejor comparación entre ellos.

7. También una propuesta en clasificación, sería modificar el algoritmo de PCA de igual manera como la propuesta NN2A, para que sea considerada la variabilidad de cada clase y no únicamente la general, para buscar un mejor clasificador basado en componentes principales.

8. Una dificultad bastante grande en NN2A y NN3A es encontrar valores parámetros que proporcionen alguna mejora a los algoritmos, por lo que se podría crear un algoritmo para su búsqueda automática.

9. En el área de series de tiempo se podrían realizar un mayor número de pruebas con diferentes archivos de pings entre las computadoras.

Como puede observarse existen muchas cosas que se pueden explorar dentro del área de componentes independientes.

Part III

APENDICES

Chapter 9

APENDICE A. ALGUNAS PROPIEDADES Y DEMOSTRACIONES MATEMATICAS

9.1 PROPIEDAD 1 (DETERMINANTE DE UNA MATRIZ)

Si $\mathbf{W}$ es una matriz cuadrada invertible de mxm cuyo determinante es denotado por $det\mathbf{W}$, entonces:

$$\frac{\delta}{\delta \mathbf{W}} \det \mathbf{W} = (\mathbf{W}^T)^{-1} \det \mathbf{W}$$

Partiendo de que la inversa de $\mathbf{W}$ se obtiene como

$$\mathbf{W}^{-1} = \frac{1}{\det \mathbf{W}} adj(\mathbf{W})$$

donde la $adj(\mathbf{W}) = \begin{pmatrix} W_{11} & \cdots & W_{n1} \\ \vdots & & \vdots \\ W_{1n} & \cdots & W_{nn} \end{pmatrix}$ donde los números escalares W_{ij} se conocen como los

cofactores. El cofactor W_{ij} se obtiene tomando primero la submatriz $(n-1)$x$(n-1)$ de $\mathbf{W}$. El determinante de $\mathbf{W}$ puede entonces ser expresado como:

$$\det \mathbf{W} = \sum_{k=1}^{n} w_{ik} W_{ij}$$

entonces

$$\frac{\delta}{\delta w_{ij}} \det \mathbf{W} = W_{ij}$$

lo cual implica directamente que:

$$\frac{\delta \det \mathbf{W}}{\delta \mathbf{W}} = adj(\mathbf{W})^T$$

pero $adj(\mathbf{W})^T$ es igual al $(\det \mathbf{W})(\mathbf{W}^T)^{-1}$,por lo que finalmente:

$$\frac{\delta |\det \mathbf{W}|}{\delta \mathbf{W}} = \frac{1}{|\det \mathbf{W}|} \frac{\delta |\det \mathbf{W}|}{\delta \mathbf{W}} = (\mathbf{W}^T)^{-1} \tag{9.1}$$

9.2 DEMOSTRACION DE APROXIMACION DE NEGENTROPIA

Partiendo del hecho de que se tienen varias esperanzas $E\{F^i(x)\}$ de diferentes funciones de x;

$$\int p(\xi) F^i(\xi) d\xi =: c_i$$

se busca la aproximación que tenga máxima entropía. Un resultado básico del método de máxima entropía[1], muestra que una densidad $p_0(\xi)$ que satisface la ecuación anterior es de la forma:

$$p_0(\xi) = A \exp(\sum_i a_i F^i(\xi))$$

que es una ecuación muy dificil de resolver, por lo que partiendo de las restricciones:

$F^{n+1}(\xi) = \xi, c_{n+1} = 0 \Longrightarrow$ media $= 0$.

$F^{n+2}(\xi) = \xi^2, c_{n+2} = 1 \Longrightarrow$varianza $= 0$.

$F^i, i = 1, \cdots, n$ como un sistema ortonormal de acuerdo a $\varphi(\xi) = \exp(-\xi^2/2)\sqrt{2\pi}$, en otras palabras:

$\int \varphi(\xi) F^i(\xi) F^j(\xi) d\xi = \left\{ \begin{array}{l} 1, \text{ si } i = j \\ 0, \text{ si } i \neq j \end{array} \right\}$, $\int \varphi(\xi) F^i(\xi) \xi^k d\xi$, para $k = 0, 1, 2$.

como $a_n <<< a_{n+2} \approx -1/2$ por ser una distribución casi gausseana(ya que ésta distribución es la que tiene mayor entropía de todas las variables con la misma varianza).

$p_0(\xi) = A \exp(\sum_i a_i F^i(\xi))$

$p_0(\xi) = A \exp(-\xi^2/2 + a_{n+1}\xi + (a_{n+2} + 1/2)\xi^2 + \sum_i a_i G^i(\xi))$

realizando la aproximación de $\exp(\epsilon) \approx 1 + \epsilon$ tenemos

$p_0(\xi) = A \exp(-\xi^2/2) \exp(a_{n+1}\xi + (a_{n+2} + 1/2)\xi^2 + \sum_i a_i G^i(\xi))$

$p_0(\xi) = A \exp(-\xi^2/2)(1 + a_{n+1}\xi + (a_{n+2} + 1/2)\xi^2 + \sum_i a_i G^i(\xi))$

$p_0(\xi) = \tilde{A} \varphi(\xi)(1 + a_{n+1}\xi + (a_{n+2} + 1/2)\xi^2 + \sum_i a_i G^i(\xi))$

donde $\tilde{A} = \sqrt{2\pi} A$. Tomando las restricciones de que la media es cero y la varianza es uno. Se obtiene una función que aproxima a $p(\xi)$:

$\hat{p}(\xi) = \varphi(\xi)(1 + \sum_i c_i G^i(\xi))$ con $c_i = E\{G^i(\xi)\}$

Calculando la entropía $H(x)$ de la función propuesta:

$$\begin{aligned}
H(x) &= -\int \hat{p}(\xi) \log \hat{p}(\xi) d\xi \\
&= -\int \varphi(\xi)(1 + \sum_i c_i G^i(\xi)) \left[\log \left(\varphi(\xi)(1 + \sum_i c_i G^i(\xi)) \right) \right] d\xi \\
&= -\int \varphi(\xi)(1 + \sum_i c_i G^i(\xi)) \left[\log(1 + \sum_i c_i G^i(\xi)) + \log \varphi(\xi) \right] d\xi \\
&= -\int (1 + \sum_i c_i G^i(\xi)) \left[\varphi(\xi) \log(1 + \sum_i c_i G^i(\xi)) + \varphi(\xi) \log \varphi(\xi) \right] d\xi \\
&= -\int \varphi(\xi) \log \varphi(\xi) d\xi - \int \varphi(\xi) \sum_i c_i G^i(\xi) \log \varphi(\xi) d\xi - \int \varphi(\xi)(1 + \sum_i c_i G^i(\xi)) \log(1 + \sum_i c_i G^i(\xi
\end{aligned}$$

como $(1 + \epsilon) \log(1 + \epsilon) = \epsilon + \epsilon^2/2 + o(\epsilon^2)$

$$
\begin{aligned}
H(x) &= H(\nu) + 0 + 0 - \int \varphi(\xi) \left[\sum_i c_i G^i(\xi) + \left(\sum_i c_i G^i(\xi) \right) /2 + o\left(\sum_i c_i G^i(\xi) \right)^2 \right] d\xi \\
&= H(\nu) - \frac{1}{2} c_i^2 \sum_i \varphi(\xi) G^i(\xi) G^i(\xi) + o\left(\left(\sum c_i \right)^2 \right) \\
&= H(\nu) - \frac{1}{2} c_i^2 + o\left(\left(\sum c_i \right)^2 \right)
\end{aligned}
$$

Despejando:

$$
H(x) - H(\nu) = J(x) \approx \frac{1}{2} \sum_{i=1}^{n} c_i^2 = \frac{1}{2} \sum_{i=1}^{n} E\{G^i(x)\}^2
$$

Si se toman dos funciones no cuadráticas G^i y G^i tal que la primera función es impar y la segunda par, se tiene la siguiente aproximación [1]:

$$
J(y) \approx k_1 \left(E\left\{ G^1(y) \right\} \right)^2 + k_2 \left(E\left\{ G^2(y) \right\} - E\left\{ G^2(\nu) \right\} \right)^2 \tag{9.2}
$$

donde k_1 y k_2 son constantes positivas y ν es una variable gausseana con media cero y varianza unitaria.

Chapter 10

APEDICE B. DEDUCCIONES DE ALGORITMOS

10.1 ALGORITMOS DE GRADIENTE

10.1.1 KURTOSIS (Suponiendo datos decorrelacionados)

Por definición:

$$
kurt(y) = E\{y^4\} - 3(E\{y^2\})^2
$$

se busca maximizar la kurtosis, por lo que se deriva con respecto al vector $\mathbf{w}$

$$
\begin{aligned}
\frac{\delta[kurt(\mathbf{w}^T\mathbf{z})]}{\delta \mathbf{w}} &= \frac{\delta}{\delta \mathbf{w}} \left[E\{(\mathbf{w}^T\mathbf{z})^4\} - 3(E\{(\mathbf{w}^T\mathbf{z})^2\})^2 \right] \\
&= \frac{\delta}{\delta \mathbf{w}} \{E\{(\mathbf{w}^T\mathbf{z})^4\} - 3\frac{\delta}{\delta \mathbf{w}}(E\{(\mathbf{w}^T\mathbf{z})^2\})^2
\end{aligned}
$$

sabemos que $E\{(\mathbf{w}^T\mathbf{z})^2\} = ||\mathbf{w}||^2$, por lo que

$$
\frac{\delta[kurt(\mathbf{w}^T\mathbf{z})]}{\delta \mathbf{w}} = \frac{\delta}{\delta \mathbf{w}} \{E\{(\mathbf{w}^T\mathbf{z})^4\} - 3\frac{\delta}{\delta \mathbf{w}}(||\mathbf{w}||^2)^2
$$

$$
\begin{aligned}
&= E\left[\frac{\delta}{\delta \mathbf{w}}\left\{(\mathbf{w}^T\mathbf{z})^4\right\}\right] - 4*3||\mathbf{w}||^3\frac{\delta}{\delta \mathbf{w}}(\mathbf{w}) \\
&= E\left(4\left(\mathbf{w}^T\mathbf{z}\right)^3\frac{\delta}{\delta \mathbf{w}}(\mathbf{w}^T\mathbf{z})\right) - 4*3||\mathbf{w}||^3 \\
&= 4\left[E\left(\mathbf{z}\left(\mathbf{w}^T\mathbf{z}\right)^3\right) - 3||\mathbf{w}||^2\mathbf{w}\right]
\end{aligned}
$$

como se desea maximizar el valor de la kurtosis y ésta puede ser negativa o positiva, se introduce un término para calcular la dirección

$$
\frac{\delta[kurt(\mathbf{w}^T\mathbf{z})]}{\delta \mathbf{w}} = 4sign\left(kurt\left(\mathbf{w}^T\mathbf{z}\right)\right)\left[E\left(\mathbf{z}\left(\mathbf{w}^T\mathbf{z}\right)^3\right) - 3||\mathbf{w}||^2\mathbf{w}\right]
$$

por lo tanto el algoritmo que se obtiene es:

$$
\Delta\mathbf{w} \propto sign\left(kurt\left(\mathbf{w}^T\mathbf{z}\right)\right)E\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\}
$$

$$
\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||
$$

10.1.2 KURTOSIS (Sin suponer datos decorrelacionados)

Por definición:

$$
kurt(y) = E\{y^4\} - 3(E\{y^2\})^2
$$

se busca maximizar la kurtosis, por lo que se deriva con respecto a el vector $\mathbf{w}$

$$
\begin{aligned}
\frac{\delta[kurt(\mathbf{w}^T\mathbf{z})]}{\delta \mathbf{w}} &= \frac{\delta}{\delta \mathbf{w}}\left[E\{(\mathbf{w}^T\mathbf{z})^4\} - 3(E\{(\mathbf{w}^T\mathbf{z})^2\})^2\right] \\
&= \frac{\delta}{\delta \mathbf{w}}\{E\{(\mathbf{w}^T\mathbf{z})^4\} - 3\frac{\delta}{\delta \mathbf{w}}(E\{(\mathbf{w}^T\mathbf{z})^2\})^2 \\
&= 4E\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\} - 6E\{(\mathbf{w}^T\mathbf{z})^2\}\frac{\delta}{\delta \mathbf{w}}E\{(\mathbf{w}^T\mathbf{z})^2\} \\
&= 4E\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\} - 12E\{(\mathbf{w}^T\mathbf{z})^2\}E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\} \\
&= 4\left[E\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\} - 3E\{(\mathbf{w}^T\mathbf{z})^2\}E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\}\right]
\end{aligned}
$$

como se desea maximizar el valor de la kurtosis y ésta puede ser negativa o positiva, se introduce un término para calcular la dirección

$$
\frac{\delta[kurt(\mathbf{w}^T\mathbf{z})]}{\delta \mathbf{w}} = 4sign\left(kurt\left(\mathbf{w}^T\mathbf{z}\right)\right)\left[E\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\} - 3E\{(\mathbf{w}^T\mathbf{z})^2\}E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\}\right]
$$

por lo tanto el algoritmo que se obtiene es:

$$
\Delta\mathbf{w} \propto sign\left(kurt\left(\mathbf{w}^T\mathbf{z}\right)\right)\left[E\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\} - 3E\{(\mathbf{w}^T\mathbf{z})^2\}E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\}\right]
$$

$$
\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||
$$

A manera de comprobación, si suponemos que se cumple que los datos están blanqueados, $E\{(\mathbf{w}^T\mathbf{z})^2\} = ||\mathbf{w}||^2$, entonces se tiene que obtener la misma fórmula que en el algoritmo anterior, así:

$$
\begin{aligned}
\frac{\delta[kurt(\mathbf{w}^T\mathbf{z})]}{\delta \mathbf{w}} &= 4sign\left(kurt\left(\mathbf{w}^T\mathbf{z}\right)\right)\left[E\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\} - 3E\{(\mathbf{w}^T\mathbf{z})^2\}E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\}\right] \\
&= 4sign\left(kurt\left(\mathbf{w}^T\mathbf{z}\right)\right)\left[E\{(\mathbf{w}^T\mathbf{z})^3\mathbf{z}\} - 3||\mathbf{w}||^2E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\}\right]
\end{aligned}
$$

por lo que solo falta comprobar que $E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\} = \mathbf{w}$, así que

$$
E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\} = \left(E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\}\right)_j = E\{\sum_{i=1}^{n}\mathbf{w}_i\mathbf{z}_i\mathbf{z}_j\} = \sum_{i=1}^{n}\mathbf{w}_iE(\mathbf{z}_i\mathbf{z}_j)
$$

donde $E(\mathbf{z}_i\mathbf{z}_j) = 0$ cuando $i \neq j$ y $E(\mathbf{z}_i\mathbf{z}_j) = 1$ cuando $i = j$, por lo que $E\{(\mathbf{w}^T\mathbf{z})\mathbf{z}\} = \mathbf{w}$.

10.1.3 NEGENTROPIA

Por definición

$$
J(\mathbf{w}^T\mathbf{z}) = \left[E\left\{G(\mathbf{w}^T\mathbf{z})\right\} - E\left\{G(\nu)\right\}\right]^2
$$

se busca maximizar la negentropia

$$\begin{aligned}\frac{\delta[J(\mathbf{w}^T\mathbf{z})]}{\delta\mathbf{w}} &\propto \frac{\delta}{\delta\mathbf{w}}\left[E\left\{G(\mathbf{w}^T\mathbf{z})\right\}-E\left\{G(\nu)\right\}\right]^2\\ &= 2\left[E\left\{G(\mathbf{w}^T\mathbf{z})\right\}-E\left\{G(\nu)\right\}\right]\frac{\delta}{\delta\mathbf{w}}\left[E\left\{G(\mathbf{w}^T\mathbf{z})\right\}-E\left\{G(\nu)\right\}\right]\end{aligned}$$

suponiendo que $\gamma = E\left\{G(\mathbf{w}^T\mathbf{z})\right\} - E\left\{G(\nu)\right\}$, y además partiendo del hecho de que $E\left\{G(\nu)\right\} = 0$ entonces

$$\begin{aligned}\frac{\delta[J(\mathbf{w}^T\mathbf{z})]}{\delta\mathbf{w}} &\propto \frac{\delta}{\delta\mathbf{w}}\left[E\left\{G(\mathbf{w}^T\mathbf{z})\right\}-E\left\{G(\nu)\right\}\right]^2\\ &= 2\gamma\frac{\delta}{\delta\mathbf{w}}\left[E\left\{G(\mathbf{w}^T\mathbf{z})\right\}-E\left\{G(\nu)\right\}\right]\\ &= \gamma E\frac{\delta}{\delta\mathbf{w}}\left\{G(\mathbf{w}^T\mathbf{z})\right\}\end{aligned}$$

ahora, sabemos que $G(\mathbf{w}^T\mathbf{z}) = g(\mathbf{w}^T\mathbf{z})$,entonces

$$\frac{\delta[J(\mathbf{w}^T\mathbf{z})]}{\delta\mathbf{w}} \propto \gamma E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\}$$

El algoritmo que se obtiene es:

$$\Delta\mathbf{w} \propto \gamma E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||$$

10.2 ALGORITMOS DE PUNTO FIJO

10.2.1 NEGENTROPÍA

Partiendo de la iteración del método de gradiente:

$$\Delta\mathbf{w} \propto \gamma E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\}$$

se puede sugerir el siguiente algoritmo de punto fijo:

$$\mathbf{w} \leftarrow E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||$$

donde la γ puede ser eliminada por la normalización. La iteración anterior no tiene buenas propiedades de convergencia, debido a que los momentos polinomiales no tienen tan buenas propiedades como las acumulables reales como la kurtosis [1]. Para mejorar dichas características, se va a modificar el algoritmo. Lo primero que se hace es mutiplicar ambos lados de la ecuación con $\alpha\mathbf{w}$, obteniendose así:

$$(1+\alpha)\mathbf{w} \leftarrow E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\} + \alpha\mathbf{w}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||$$

lo cual tiene el mismo punto fijo. Entonces se necesita una α, tal que mejore el algoritmo. Ésto se puede hacer por medio del método de Newton, ya que al aplicarlo al método del gradiente, generalmente representa una mejora en la convergencia [1]. El problema de utilizar este método es que requiere inversiones de matrices en cada paso, por lo que no representa una mejora en el tiempo. Utilizando las propiedades de ICA, se puede llegar a una aproximación del método de Newton sin necesidad de calcular las inversas.

La derivación del método es la siguiente: Partiendo de que el máximo de la negentropia de $\mathbf{w}^T\mathbf{z}$(utilizando su aproximación) se tiene en donde exista un optimo en $E\left\{G(\mathbf{w}^T\mathbf{z})\right\}$.Esto implica que el óptimo de $E\left\{G(\mathbf{w}^T\mathbf{z})\right\}$ se encuentra en donde el Lagrangiano es cero.

$$F = E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\} + \beta\mathbf{w} = 0$$

así,

$$\frac{\delta F}{\delta \mathbf{w}} = E\left\{\mathbf{z}\mathbf{z}^T g(\mathbf{w}^T\mathbf{z})\right\} + \beta\mathbf{I} = 0$$

como los datos están blanqueados, se puede realizar la siguiente aproximación:

$$E\left\{\mathbf{z}\mathbf{z}^T g(\mathbf{w}^T\mathbf{z})\right\} \approx E\{\mathbf{z}\mathbf{z}^T\}E\{g(\mathbf{w}^T\mathbf{z})\} + \beta\mathbf{I} \approx E\{g(\mathbf{w}^T\mathbf{z})\mathbf{I}$$

entonces, partiendo de que el método de Newton es:

$$f(x+\epsilon) \approx f(x) + f'(x)\epsilon$$

$$\epsilon_0 = -\frac{f(x_0)}{f'(x_0)}$$

se tiene:

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\right\} + \beta\mathbf{w}}{E\{g'(\mathbf{w}^T\mathbf{z})\} + \beta}$$

y multiplicando por $E\{g'(\mathbf{w}^T\mathbf{z})\}+\beta$ se tiene el siguiente algoritmo:

$$\mathbf{w} \leftarrow E\left\{\mathbf{z}g(\mathbf{w}^T\mathbf{z}) - E\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w}\right\}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||$$

Chapter 11

APENDICE C. MODELO ICA CON RUIDO

11.1 INTRODUCCION AL MODELO

En cualquier medición se tiene ruido presente, por lo que, un modelo de ICA más real, contempla un término que representa el ruido que se asume como de tipo aditivo y se expresa de la siguiente manera:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \tag{11.1}$$

donde $\mathbf{n} = (n_1, ...n_n)$ es el vector que lo representa. Generalmente se supone que:

1. El ruido es independiente de los componentes.
2. El rudo es gauseano

La matriz de covarianza del ruido, $\mathbf{\Sigma}$, se asume que tiene la forma $\sigma^2\mathbf{I}$ y se tienen las mismas suposiciones que en el modelo sin ruido(independencia y no gauseanidad).

En el caso básico en donde la matriz de covarianza tiene la forma $\sigma^2\mathbf{I}$ el ruido es considerado como un ruido sensado, ya que las señales de ruido son sumadas separadamente en cada sensor. Las fuentes de ruido pueden ser modeladas con una ecuación ligeramente diferente que la anterior

$$\mathbf{x} = \mathbf{A}(\mathbf{s} + \mathbf{n}) \tag{11.2}$$

donde otra vez la matriz de covarianza del ruido es diagonal. Si se consideran los componentes independientes con ruido, dados por $\tilde{s}_i = s_i + n_i$, el modelo se puede reescribir como:

$$\mathbf{x} = \mathbf{A}\tilde{\mathbf{s}} \tag{11.3}$$

que es el modelo de ICA básico, pero con componentes independientes modificados, donde los componentes de $\tilde{\mathbf{s}}$ cumplen que son no gausseanos e independientes, por lo que se pueden estimar por un método básico de ICA. De ésta manera, se tendrían que estimar la matriz de mezclas y los componentes independientes con ruido. Un problema adicional es calcular los componentes independientes originales tomando como base los componentes con ruido.

Asumiendo que la matriz de covarianza del ruido tiene la forma:

$$\sum = \mathbf{A}\mathbf{A}^T\sigma^2 \tag{11.4}$$

entonces, el vector de ruido puede ser transformado en otro vector $\tilde{\mathbf{n}} = \mathbf{A}^{-1}\mathbf{n}$, el cual es una fuente equivalente de ruido. Entonces, la ecuación 11.1 original será ahora:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{A}\tilde{\mathbf{n}} = \mathbf{A}(\mathbf{s} + \tilde{\mathbf{n}}) \tag{11.5}$$

de esta manera, la covarianza de $\tilde{\mathbf{n}}$ es $\sigma^2\mathbf{I}$, y entonces los componentes transformados en $\mathbf{s} + \tilde{\mathbf{n}}$ son independientes y se puede calcula la matriz de mezclas $\mathbf{A}$ por medio de métodos básicos de ICA

11.2 ESTIMACIÓN DE LA MATRIZ DE MEZCLAS

No existen muchos métodos de ICA con ruido, pero algunas maneras de calcular la matriz de mezclas son las siguientes:

11.2.1 Técnicas de bias removal

Probablemente ésta sea la mejor aproximación, esta técnica implica que los métodos de ICA ordinarios son modificados tal que el sesgamiento hecho por el ruido sea reducido. El modelo libre de ruido como sabemos es de la forma

$$\mathbf{v} = \mathbf{A}\mathbf{s} \tag{11.6}$$

La idea básica es encontrar proyecciones, $\mathbf{w}^T\mathbf{v}$, en las cuales la no gauseanidad, sea maximizada localmente para los datos blanqueados, con la restriccion de $||\mathbf{w}|| = 1$. Se tiene $\mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{v} + \mathbf{w}^T\mathbf{n}$, pro lo que la meta es encontrar la medida de no gausseanidad $\mathbf{w}^T\mathbf{v}$ de los datos observados $\mathbf{w}^T\mathbf{x}$ de tal manera que la medida no esté afectada por el ruido $\mathbf{w}^T\mathbf{n}$.

Una manera de realizar lo anterior es por medio de la kurtosis, ya que esta medida no es afectada por el ruido gausseanao. $\mathbf{w}^T\mathbf{v} = \mathbf{w}^T\mathbf{v}$. Denotando la matriz de covarianza de los datos con ruido como $\mathbf{C} = \mathbf{E}\{\mathbf{x}\mathbf{x}^T\}$, el blanqueado ordinario se reemplaza por CHECAR PORQUE

$$\tilde{\mathbf{x}} = (\mathbf{C} - \mathbf{\Sigma})^{-1/2}\mathbf{x} \tag{11.7}$$

a ésta operación se le conoce como "*cuasi-blanqueado*". Después de esta operación el modelo de datos quasi-blanqueados $\mathbf{x}$ es:

$$\tilde{\mathbf{x}} = \mathbf{B}\mathbf{s} + \tilde{\mathbf{n}}$$

donde $\mathbf{B}$ es ortogonal, y $\tilde{\mathbf{n}}$ es una transformación lineal del ruido original en 11.1.

Otra manera de obtener las proyecciones es utiliazando la definición de negentropia $J_G(\mathbf{w}^T\mathbf{v}) = [E\{G(\mathbf{w}^T$ donde G es una función suficientemente regular y no cuadrática y ν es una variable estándar. Se puede calcular $J_G(\mathbf{w}^T\mathbf{v})$ del modelo de datos libre de ruido. Denotando z como una variable no gausseana, y por n una variable de ruido gauseano de varianza σ^2, se puede expresar la relación entre $E\{G(z)\}$ y $E\{G(z+n)\}$, pero es sumamente complicado [1].

11.2.2 FastICA para variables con ruido

Utiilzando las medidas anteriores, se puede derivar una variante del algoritmo de FastICA. El algoritmo tiene la forma: CHECAR DEDUCCION

$$\mathbf{w}^*=E\left\{\tilde{\mathbf{x}}g(\mathbf{w}^T\tilde{\mathbf{x}})\right\}-(\mathbf{I}+\tilde{\mathbf{\Sigma}})\mathbf{w}E\{g'(\mathbf{w}^T\tilde{\mathbf{x}})\} \tag{11.8}$$

donde $\mathbf{w}^*$, el nuevo valor de $\mathbf{w}$, es normalizado a norma unitaria después de cada iteración, y $\tilde{\mathbf{\Sigma}}$ está dada por:

$$\tilde{\mathbf{\Sigma}} = E\{\tilde{\mathbf{n}}\tilde{\mathbf{n}}^T\} = (\mathbf{C}-\mathbf{\Sigma})^{-1/2}\mathbf{\Sigma}(\mathbf{C}-\mathbf{\Sigma})^{-1/2} \tag{11.9}$$

La función g es la derivada de G, y es:

$$g_1(u) = \tanh(u), g_2(u) = u\exp(-u^2/2), g_3(u) = u^3 \tag{11.10}$$

11.3 ESTIMACION DE LOS COMPONENTES INDEPENDIENTES LIBRES DE RUIDO

11.3.1 Estimación máxima a posteriori

En ICA con ruido, no es suficiente estimar la matriz de mezclas. Invirtiendola, se obtiene:

$$\mathbf{Wx} = \mathbf{s} + \mathbf{Wn} \tag{11.11}$$

en otras palabras, solo de tiene la estimación de los componentes con ruido. Para estimar los c.i. libres de ruido, se puede partir de la estimación maxima a posteriori (MAP). Es decir, se toman los componentes $\tilde{s}_i$ que maximicen la verosimilitud, lo cual se conoce como estimador de máxima verosimilitud. Para calcular el estimador MAP, se toma el gradiente de la log-verosimilitud con respecto a $\mathbf{s}(t), t = 1, ...T$, y se iguala a 0; donde

$$\log L(\mathbf{A}, \mathbf{s}(1), ..., \mathbf{s}(T)) = -\sum_{t=1}^{T}\left[\frac{1}{2}\|\mathbf{As}(t)-\mathbf{x}(t)\|^2_{\Sigma^{-1}} + \sum_{i=1}^{n} f_i(s_i(t))\right] + C \tag{11.12}$$

obteniendose:CHECAR DEMO

$$\hat{\mathbf{A}}^T\mathbf{\Sigma}^{-1}\hat{\mathbf{A}}\hat{\mathbf{s}}(t) - \hat{\mathbf{A}}^T\mathbf{\Sigma}^{-1}\mathbf{x}(t) + f'(\hat{\mathbf{s}}(t)) = 0 \tag{11.13}$$

donde f' se aplica separadamente a cada componente del vector $\hat{\mathbf{s}}(t)$

Chapter 12

APENDICE D. MÉTODOS DE CLASIFICACIÓN

12.1 METODOS PROTOTIPO

Teniendo un conjunto de datos de entrenamiento que consiste en N pares $(x_1, g_1) \cdots (x_n, g_N)$ donde g_i es un identificador de clase que toma valores entre $\{1, 2, \ldots K\}$. Los métodos de prototipo representan al conjunto de entrenamiento como puntos en el espacio característico. Cada protipo es asociado a un identificador de clase y cada clasificación de un punto x se hace a la clase del prototipo más cercano. Ésta distancia de cercanía generalmente se toma en base a la distancia euclidiana; cada característica del conjunto de entrenamiento es previamente estandarizada (con media 0 y varianza 1). Éste tipo de método es efectivo si se tienen bien posicionados los prototipos, aunque existen otros que difieren en la forma en la que se toman.

12.1.1 K-MEDIAS CLUSTERING

K-Medias es uno de los muchos algoritmos desarrollados para realizar una clasificación no supervisada de elementos. El usuario escoge el numero de clusters, R, y el algoritmo produce iterativamente un conjunto de agrupamientos de los datos con R centros; puede describirse en dos pasos:

- Para cada centro, identificar el subconjunto de puntos de entrenamiento que es más cercano a él que a cualquier otro centro.
- Calcular la media de cada conjunto y hacerla el centro del cluster.

Ambos pasos son iterados hasta convergencia. Inicialmente se escogen aleatoriamente los centros de las clases; para utilizar éste algoritmo como clasificador se siguen los siguientes pasos:

- Aplicar el método de K-medias para entrenar datos en cada clase separadamente, usando R prototipos por clase.
- Asignar un identificador de clase a cada uno de los KxR prototipos.
- Clasificar una nueva característica x de la clase al prototipo más cercano.

12.1.2 MEZCLAS GAUSSEANAS

El modelo de mezclas gausseanas es muy parecido al de K-medias.Cada cluster se describe como una densidad gausseana, con un centroide y una matriz de covarianza. Este método consta de dos pasos:

- En el paso E, cada observación es asignada con un peso a cada cluster basándose en la verosimilitud entre el cluster y la gausseana. A las observaciones más cercanas al centro del cluster se les asigna un peso de 1 y a las demás un peso de 0; a las que están igual de cerca de dos clusters, se les asigna un peso proporcional a la distancia entre ambos.
- En el paso M, cada observación contribuye a la media (y covarianza) de cada cluster.

A este método se le conoce como un método ligero de clusterización, mientras que al algoritmo de K-medias como pesado.

12.2 CLASIFICADORES DE K-VECINOS MAS CERCANOS

Éste método está basado en memoria; dado un punto de búsqueda x_0, se encuentran k puntos de entrenamiento $x_{(r)}, r = 1, ..., k$ más cercanos en distancia a x_0(generalmente distancia euclidiana), y se clasifican utilizando una mayoría en la votación entre los k-vecinos. Los datos

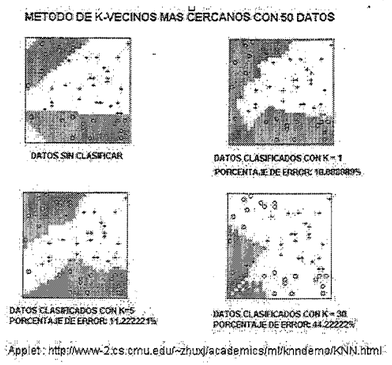


Figure 12-1:

son previamente estandarizados, para evitar problemas de unidades.La fig. muestra algunos ejemplos de clasificación y sus porcentajes de error.

Además de su simplicidad, K-vecinos más cercanos tienen gran éxito en diferentes conjuntos de datos como son los dígitos escritos a mano y las imágenes de satélite. Es normal que las fronteras de desición sean muy irregulares, ya que comunmente cada clase puede tener varios prototipos. En la clasificación de 1-vecino mas cercano, cada punto de entrenamiento es un prototipo.

El clasificador de Bayes dice que los datos se clasifican utilizando la clase más probable, utilizando la distribución condicional discreta $\Pr(G|X)$. La tasa de error el clasificador de Bayes se conoce como tasa de Bayes.Un resultado importante muestra que asintóticamente la tasa de error del clasificador del vecino más cercano nunca es mayor que dos veces la tasa de Bayes (Cover y Hart, 1967).

Error de Bayes: $1 - p_{k^*}(x)$

Error de 1-vecino más cercano: $\sum_{k=1}^{K} p_k(x)(1 - p_k(x))$,

$\geq 1 - p_{k^*}(x)$

12.3 PROJECTION PURSUIT COMO APRENDIZAJE SUPERVISADO

Teniendo un problema de aprendizaje supervisado, asumamos que tenemos un vector de entrada X con p componentes y un objetivo Y. Sea w_m, $m = 1, 2, ..., M$, p-vectores de parámetros desconocidos. El modelo de regresión de proyection pursuit (PPR) tiene la forma:

$$f(X) = \sum_{m=1}^{M} g_m(w_m^T X)$$

Este es un modelo aditivo, pero en las características derivadas $V_m = w_m^T X$ más bien que las entradas.Las funciones g_m no están especificadas y son estimadas a lo largo de las direcciones w_m por medio algunos métodos suaves. La función $g_m(w_m^T X)$ es llamada función cordillera en R^p y varía solo en las direcciones definidas por el vector w_m. La variable escalar V_m es la proyección de X en el vector unitario w_m, se puede ver que el modelo se ajusta bien, de ahí el nombre de "proyection pursuit". El modelo PPR es muy general, ya que sorprendentemente puede aproximar una gran cantidad de clases de modelos.De hecho, si M es lo suficientemente grande, para una selección apropiada de g_m, se puede aproximar cualquier función continua en R^p. A este tipo de modelos se les conoce como aproximadores universales, lo malo es que a mayor escala, se vuelve muy compleja su interpretación. De esta manera, el PPR es bueno para predicción, pero no para generar un modelo de interpretación. El modelo con $M = 1$ es la excepción, ya que es más general que el modelo de regresión y ofrece una interpretación similar.

Los minimizadores aproximados de la función de error son:

$$\sum_{i=1}^{N} \left[y_i - \sum_{m=1}^{M} g_m \left(w_m^T x_i \right) \right]^2$$

sobre las funciones g_m y los vectores de dirección $w_m, m = 1, 2, ...M$. Considerando un solo término ($M = 1$); dada ladirección del vector inicial w, se forman las variables derivadas $v_i = w^T x_i$. por lo que tenemos un problema suave en una dimensión y podemos aplicar cualquier método de dispersión (scatterplot) para estimar g. Ahora, desde otro punto de vista, dado un vector inicial g, se desea minimizar la ecuación anterior sobre w. Se puede utilizar una búsqueda

de Gauss-Newton entonces:

$$g(w^T x_i) \simeq g(w_{vieja}^T x_i) + g'(w_{vieja}^T x_i)(w - w_{vieja})^T x_i$$

que es

$$\sum_{i=1}^{N} \left[y_i - g(w^T x_i)\right]^2 \simeq \sum_{i=1}^{N} g'(w_{vieja}^T x_i)^2 \left[\left(w_{vieja}^T x_i + \frac{y_i - g(w_{viejo}^T x_i)}{g'(w_{vieja}^T x_i)}\right) - w^T x_i\right]^2$$

Para minimizar el lado derecho, se hace una regresión de minims cuadrados con objetivo $w_{vieja}^T x_i + (y_i - g(w_{viejo}^T x_i))/g'(w_{vieja}^T x_i)$ de la entrada x_i, con pesos $g'(w_{vieja}^T x_i)^2$y sin ningún término de intersección. Esto produce el nuevo vector de coeficientes w_{nuevo}

Estos dos pasos, la estimación de g y de w, son iterados hasta la convergencia. Cuando se tienen más término en el modelo PPR, al modelo anterior se le añade un par (w_m, g_m) en cada etapa.

12.4 REDES NEURONALES

El término de redes neuronales se aplica a diversas clases de modelos y de métodos de aprendizaje. El método descrito a continuación es conocido como perceptron con una sola capa. Una red neuronal es un modelo de clasificación, tipicamente representado por un diagrama como el de la fig. ; se utiliza tanto para clasificación como para regresión. Para la segunda tipicamente $K = 1$ y solo hay una unidad de salida Y_1, para la clasificación de $K-$clases, se hay K unidades hasta "arriba" donde la k-ésima unidad modela la probabilidad de la clase k. Hay K medidas de objetivos Y_k, $k = 1...K$, y cada uno se codifica como una variable que toma los valores de 0 o 1. Las características secundarias Z_m son combinaciones lineales de las entradas, y cada medida Y_k está modelada como combinaciones lineales de Z_m,

$Z_m = \sigma\left(\alpha_{0m} + \alpha_m^T X\right), m = 1, ...M$

$T_k = \beta_{0k} + \beta_k^T Z, k = 1, ..., K$

$f_k(X) = g_k(T), k = 1, ..., K$

donde $Z = (Z_1, Z_2, ...Z_M)$,y $T = (T_1, T_2, ..., T_K)$

la función de activación $\sigma(v)$ es usualmente una sigmoide (Fig.) $\sigma(v) = 1/(1 + e^{-v})$,

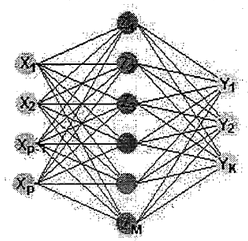


Figure 12-2:

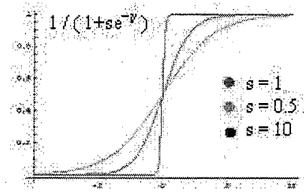


Figure 12-3:

aunque a veces se utilizan funciones de base radial.

La función de salida $g_k(T)$ permite una transformación final del vector de salidas T. Para regresión se escoge típicamente la función identidad $g_k(T) = T_k$ o la función softmax $g_k(T) = \frac{e^{T_k}}{\sum_{l=1}^{K} e^{T_l}}$.

Las unidades que se encuentran a la mitad de la red, calculan las características derivadas Z_m y son conocidas como capas ocultas, ya que los valores de Z_m no son observads directamente. Si σ es la función identidad, entonces todo el modelo es un modelo lineal. Si se introduce una transfomración no lineal σ, amplia grandemente las clases de los modelos lineales. El promedio de activación de la sigmoide, depende de la norma de α_m,y si $||\alpha_m||$ es muy pequeña, la unidad opera como una parte lineal.

Hay que notar que el modelo de red neuronal con una sola cada oculta tiene exactamente la misma forma que el modelo de projection pursuit descrito anteriormente. La diferencia es que

el modelo PPR utiliza funciones no paramétricas $g_m(\nu)$, mientras que la red neuronal utiliza un función simple basada en $\sigma(\nu)$, con tres parámetros libres en cada argumento. En detalle, viendo un modelo de red neuronal, se puede identificar:

$$g_m(w_m^T X) = \beta_m \sigma(\alpha_{0m} + \alpha_m^T X) = \beta_m \sigma(\alpha_{0m} + ||\alpha_m||(w_m^T X))$$

donde $w_m = \alpha_m/||\alpha_m||$ es el m-ésimo vector unitario. Como $\sigma_{\beta,\alpha_0,s}(\nu) = \beta\sigma(\alpha_0 + s\nu)$ tiene menor complejidad que un $g(\nu)$ general mas no paramétrico. Debido a esto, no es sorprendente que una red neuronal utilice 20 o 100 funciones, mientras que el modelo PPR tipicamente utiliza menor número de términos (M=5 o 10).

Finalmente, el nombre de "red neuronal" se deriva del hecho de que fueron diseñados primeramente basados en el cerebro humano. Cada unidad representa una neurona, y sus conecciones representan las sinapsis. En modelos anteriores, las neuronas eran disparadas cuando se excedia un cierto umbral. en el modelo descrito anteriormente, ésto corresponde a utilizar una función por pasos para $\sigma(Z)$ y $g_m(T)$.

Chapter 13

APENDICE E. CARACTERISTICAS DE LOS DATOS DE PRUEBA

13.0.1 DESCRIPCION DE ATRIBUTOS DE SEGMENT.DAT

Cada observación de esta base de datos representa una región de 3x3 pixeles por medio de 19 atributos continuos descritos a continuación:

1. region-centroide-col: La columna del pixel central de la región.
2. región-centroide-ren: El renglón del pixel central de la región.
3. región-pixel-contador: El número de pixeles en la región = 9.
4. densidad-línea-corta-5: El resultado del algoritmo de extracción que cuenta cuántas líneas de longitud 5 (en cualquier orientación) con bajo contraste, menor o igual a 5, pasan a través de la región.
5. densidad-línea-corta-2: Lo mismo que la anterior pero cuenta las lineas con alto contraste, mayor que 5.
6. media-vborde: Medida del contraste de los pixeles horizontalmente adyacentes en la región. Hay 6, la media y la desviación estándar están dadas. Este atributo es utilizado

como un detector de bordes verticales.

7. sd-vborde (ver 6)

8. media-hborde: Medida del contraste entre los pixeles adyacentes verticalmente utilizado como detector de líneas horizontales.

9. sd-hborde (ver 8).

10. intensidad-media: El promedio en la región de (R + G+ B)/3

11. rawrojo-media: El promedio en la región del valor R.

12. rawazul-media: El promedio en la región del valor B.

13. rawverde-media: El promedio en la región del valor G.

14. exrojo-media: Medida del exceso en rojo: (2R - (G + B))

15. exazul-media: Medida del exceso en azul: (2B - (G + R))

16. exverde-media: Medida del exceso en verde: (2G - (R + B))

17. valor-media: Tranformación no lineal en 3D del RGB (El algoritmo puede ser encontrado en Foley and VanDam, Fundamentals of Interactive Computer Graphics)

18. saturación-media: (ver 17)

19. matiz-media: (ver 17).

13.0.2 DESCRIPCION DE ATRIBUTOS DE DIABETES.DAT

A continuación se describen los atributos que componen cada renglón de la base de datos:

1. Número de embarazos.

2. Concentración de glucosa dos horas después de la prueba oral de tolerancia a la glucosa

3. Presión sanguínea (mm Hg).

4. Grosor de los dobleces de la piel en los triceps (mm).

5. Suero de insulina por 2 horas (mu U/ml).

6. Indice de masa corporal (peso en kg/(estatura en metros)^2).

7. Función de pedigrí de diabetes.

8. Edad (años).

9. Variable de clase (0 ó 1).

10. Valores de atributos perdidos: Ninguno.

11. Distribución de clases (valor de clase 1 es interpretado como "prueba positiva para diabetes".

12. Número de clases: 0 (500), 1 (268).

13. Análisis estadístico breve:

Atributo	Media	Desv.Std.
1	3.8	3.4
2	120.9	32.0
3	69.1	19.4
4	20.5	16.0
5	79.8	115.2
6	32.0	7.9
7	0.5	0.3
8	33.2	11.8

Bibliography

[1] Aapo Hyvärinen, Juha Karhunen, Erkki Oja. *Independent Component Analysis.* John Wiley & Sons, Inc.

[2] Trevor Hastie,Robert Tibshirani,Jerome Friedman. *The Elements of Statistical Learning:Data Mining, Inference, and Prediction.* Springer.

[3] Miguel Ángel Carreira-Perpiñán. *Continuous latent variable models for dimensionality reduction and sequencial data reconstruction.* Departament of Computer Science University of Sheffield, UK.Febrero 2001

[4] Matthew Partridge and Rafael Calvo. *Fast Dimensionality Reduction and Simple PCA.* SEDAL, Departament od Electrical Engineeing University of Sydney, Australia. December 2, 1997

[5] R.D. King, C. Feng y A. Sutherland *StatLog: Comparison of Classification Algorithms on Large Real-World Problems.*

[6] Aapo Hyvarinen y Erkki Oja. *Independent Component Análisis: A tutorial.* Universidad Tecnológica de Helsinki, Finlandia.

[7] Aapo Hyvarinen. *Survey on Independent Component Analysis.* Universidad Tecnológica de Helsinki, Finlandia

[8] Miguel Á. Carreira-Perpiñán. *A Review of Dimension Reduction Techniques.* Departamento de Ciencias de la Computación, Universidad de Sheffield. Enero 27, 1997

[9] V.David Sánchez A. *PP, ICA y PCA en BSS.* Advanced Computational Intelligent Systems. Pasadena CA.

[10] A. Hyvarinen, P.O.Hoyer, y E.Oja. *Image denoising by sparce code shrinkage.* In S. Haykin and B.Kosko editors. Intelligent Signal Processing. IEEE Press, 2001.

[11] William H. Press, Saul A. Teukolsky, Willian T. Vetterling, Brian P. Flannery. *Numerical Recipies in C++.* Cambridge University Press.