Desarrollo de un CDSS basado en datos biomédicos: aplicación a diagnóstico mediante cardiotocografías prenatales

|  |  |  |
| --- | --- | --- |
| I N F O R M A C I Ó N  *Keywords:*  CDSS, outliers, PCA, filters, wrappers, modelo de clasificación, LDA, QDA, K-vecinos, Holdout, K-Fold, Leave-one-out |  | A B S T R A C T  La cardiotocografía (CTG) prenatal es una prueba de detección utilizada en el embarazo para identificar fetos con riesgo de desarrollar hipoxia a partir de registro simultáneos la frecuencia cardíaca fetal, los movimientos fetales y las contracciones uterinas. Mediante el uso de Sistemas Ayuda a la Decisión Médica (CDSS) buscamos prevenir errores médicos y servir como soporte al diagnóstico y pronóstico de esta técnica de registro. Para ello hemos desarrollado y entrando un modelo que nos permita predecir el estado del feto a partir de las 21 variables proporcionadas por el CTG de una muestra de 440 pacientes. El procedimiento ha sido realizar un análisis exploratorio de los datos solucionando outliers y datos perdidos de la muestra, a continuación, llevar a cabo una extracción de las características para seleccionar las variables más relevantes y obviar las menos representativcas. Así, por una parte, aplicamos aproximación indirecta o filters con los métodos *relieff* y *fscore* quedándonos con 16 variablesy por otra parte realizaremosaproximación directa o *wrappers*, con el método *forward* quedándonos con 12 variables. Seguidamente se realiza *PCA* con las 16 variables y otro con las 12. Para cada una de las anteriores reducciones de dimensionalidad se evaluaron distintos modelos que permitieran establecer una clasificación entre los tres posibles estados fetales, obteniendo que los modelos basado en memoria K-vecinos con parámetros de vecindad uno era el que mejor se ajustaba y presentaba menor tasa de error. Posteriormente se entrenaron y evaluaron estos modelos con los métodos *HoldOut* (35%), *K-Fold* (K=15) y *Leave-One-Out*. Una vez que tengamos los datos de validación volveremos a ejecutar el modelo predictivo para ver la tasa de error resultante.  . |

1. Introducción

Los CDSS (Clinical Decission Support System) son sistemas computacionales diseñados para ayudar a la toma de decisiones clínica basándose en datos de pacientes individuales para generar evaluaciones de pacientes específicos o recomendaciones para los clínicos. Dicho de otra forma, los CDSS aportan conocimiento específico mediante el análisis de múltiples variables para constituir una herramienta de valor en la toma de decisiones clínica, principalmente en cuanto a diagnóstico y pronóstico, ofreciendo probabilidades o riesgos asociados a cada posible diagnóstico utilizando datos y/o

señales biomédicas.[1]

La cardiotocografía prenatal es el registro continuo de la frecuencia cardíaca fetal que se obtiene mediante un transductor de ultrasonido colocado sobre el abdomen materno. Esta técnica se usa ampliamente en el embarazo como un método de evaluación del bienestar fetal, predominantemente en los embarazos con mayor riesgo de complicaciones, ya sea causados por trastornos médicos de la madre o del propio feto.[2]

La aplicación de metodologías y conceptos propios del aprendizaje automático pueden con datos obtenidos a partir de cardiotocografía prenatal puede usarse para crear un CDSS que prediga el estado de salud del feto a partir de esta señal biomédica.

1. Materiales

Nuestro CDSS basa su conocimiento en una compilación de 440 cardiotocogramas fetales de los cuales se extrajeron un total de 21 características diagnósticas: latidos por minuto (LB), número de aceleraciones por segundo (AC), número de movimientos fetales por segundos (FM), número de contracciones uterinas por segundo (CU), número de desaceleraciones de la luz por segundo (DL), número de deceleraciones severas por segundo (DS), número de deceleraciones prolongadas por segundo (DP), porcentaje de tiempo con variabilidad anormal a corto plazo (ASTV), valor medio de la variabilidad a corto plazo (MSTV), porcentaje de tiempo con variabilidad anormal a corto plazo (ALTV), valor medio de la variabilidad a largo plazo (MLTV), anchura del histograma de FHR (Anchura), mínimo de histograma de la FCF (Mínimo), máximo de histograma de FHR (Máximo), número de picos de histograma (Nmax), número de ceros del histograma (Nzeros), modo histograma (Modo), media del histograma (Media), mediana del histograma (Mediana), desviación de histograma (Desviación), tendencia del histograma (Tendencia).

Estos datos, recogidos en el fichero *Data\_train.mat* fueron clasificados por tres expertos en tres posibles estados fetales: Normal (N), Sospechoso (S) y Patológico (P). *Data\_train.mat* será usado para crear y entrenar nuestro modelo y, posteriormente se usará otro fichero de datos (DATA\_TEST.mat) para validar nuestro CDSS.

1. Métodos

Una vez estamos en disposición de nuestro *dataset* con las variables de análisis y la muestra poblacional seleccionada, nuestro CDSS debe seguir una serie de etapas para ser capaz de desarrollar bien su cometido: Preproceso, Transformación y Aprendizaje. En primer lugar, se debe realizar un Preproceso mediante un análisis exploratorio de los datos para estudiar la calidad de estos. A continuación, se realiza la Transformación a base de extraer características para reducir la dimensionalidad. Esto nos permitirá crear modelos robustos y sencillos de clasificación en la etapa de Aprendizaje, la cual finalizaremos con la evaluación de los modelos obtenidos tanto para los datos de entrenamiento como para los de validación.[3]

* 1. Análisis exploratorio

Nuestro código, en primer lugar, busca datos perdidos y *outliers* y trata estos problemas. El tratamiento de los datos perdidos se basa en rellenar con la media de toda la variable, de la misma manera, los outliers se resuelven con la mediana de toda la variable.

A continuación, se realiza un estudio gráfico y numérico de la naturaleza y relación entre las variables que tenemos. Esto lo llevamos a cabo con histogramas, función densidad de probabilidad, gráficos cuantil-cuantil, y gráficos de correlación para cada variable y distinguiendo las tres clases de datos que existen (N, S, P). [3]

* 1. Extracción de características

Para seleccionar las variables más relevantes usaremos, en primer lugar, la aproximación indirecta o *filters*. Usamos tanto el método *Fscore* como *relieff*. Una vez eliminadas las variables menos significativas obtenidas con estos métodos, extraemos características usando el Análisis de Componentes Principales (PCA) queriendo representar el 95% de la varianza de los datos. Por otro lado, tras realizar *fscore* y *relieff* podemos reducir aún más la dimensionalidad usando la aproximación directa o *wrappers*, en concreto, con el método forward; y, de nuevo, realizar PCA con esta nueva reducción

De esta manera, tenemos 4 reducciones de dimensionalidad: una únicamente quitando las variables menos significativas con *Filters*, otra haciendo PCA a estos datos, otra realizando los métodos *Wrappers* a los datos obtenidos con *Filters* y la última realizando PCA a los datos obtenidos con *Wrappers*.

* 1. Clasificación

Para cada una de las anteriores reducciones de dimensionalidad se evaluaron distintos modelos que permitieran establecer una clasificación entre los tres posibles estados fetales, dos de tipo generativo: Análisis de Discriminante Lineal (LDA) y Análisis de Discriminante Cuadrático (QDA), aunque este último solo pudo evaluarse para las reducciones de dimensionalidad que habían usado PCA y no para las otras dos; y un modelo basado en memoria: el modelo K-vecinos (K-vecinos) con 3 parámetros de vecindad diferentes (, y ). [4]

Así que tenemos 5 modelos diferentes para cada una de las 4 reducciones de dimensionalidad anteriormente realizadas, a excepción de los modelos QDA de la reducción con Filters y de la reducción con Wrappers, es decir, un total de 18 modelos. Con el propio dataset inicial podemos validar estos modelos y obtener el error que cometen al intentar clasificar estos datos de validación con las fronteras establecidas con los datos de entrenamiento. De esta manera podemos elegir, de entre los 18 modelos realizados los 2 que presenten un menor error.

* 1. Evaluación de los modelos

Tras elegir los 2 mejores modelos que mejores se ajustan a nuestros datos de entrenamiento (*Data\_train.mat*), debemos entrenar y evaluar estos modelos para obtener el que presente una menor tasa de error. Esto se ha realizado dividiendo todo el *Data\_train.mat* en un conjunto de entrenamiento y un conjunto de test para estimar el ratio de error del modelo ya entrenado. Los entrenamientos y evaluaciones se han conseguido a través del método *Holdout* con un 35% de datos como test y el resto como entrenamiento; el método *K-Fold* dividiendo en k=15 subconjuntos, de manera que se aplica el método *Holdout* k veces, usando cada vez un subconjunto distinto para validar el modelo entrenado con los k-1 subconjuntos restantes; y el método *Leave-One-Out*, equivalente al *K-Fold* pero con un parámetro k igual al número total de ejemplos en el conjunto de datos.[5][6]

Se elegirá modelo final aquel que presente una menor tasa de error tras ser entrenado y validado con los datos iniciales *Data\_train.mat*.

* 1. Introducción de datos de validación de nuestro CDSS

Tras elegir nuestro modelo debemos validarlo con nuevos datos (*Data\_test.mat*) para comprobar el sobreajuste que haya podido realizar con los datos de entrenamiento (*Data\_train.mat*). Para esto, prepararemos estos nuevos datos tratándolos de igual manera que a aquellos con los que se ha construido el modelo, es decir, se identificarán y solventarán los datos ausentes y los *outliers*, se reducirá la dimensionalidad de la misma manera que al modelo finalmente elegido, se aplicará el mismo modelo de clasificación y, finalmente, se le aplicará el modelo construido y se comprobará la tasa de error que se obtiene al aplicarlo a estos datos.

1. Resultados

En un primer análisis de los datos originales recogidos en *Data\_train.mat* podemos seleccionar características mediante la aproximación indirecta o filters con los métodos *relieff* y *fscore,* obteniéndose las variables de mayor a menor importancia para cada uno de los métodos anteriores, podemos ver el grado de coincidencia de ambos resultados para las últimas cinco variables. Además, si evaluamos sus respectivas varianzas tienen un valor de cero, presentando por tanto una distribución uniforme. Y ante la posibilidad de que pueda existir correlación entre las variables restantes recurrimos al gráfico de correlación, que representa a través de colores la correlación entre más variables aparte de las ya eliminadas y que es lo que nos queda por seguir analizando. Pero hasta el momento podemos concluir que las variables *AC*, *DL*, *DS*, *DP* y *Nzeros* son totalmente descartables. La nueva matriz queda entonces formada por 16 variables.

En un segundo análisis de las variables seguimos seleccionando características mediante una aproximación directa o *wrappers*, con el método *forward*. Este método nos devuelve mediante 0s y 1s las variables que obtienen directamente los mejores resultados para el modelo predictivo. Quedándonos por tanto con cuatro variables menos: *MSTV*, *Width*, *Min*, *Median*. Resultando al final una matriz formada por 12 variables.

A continuación, se realiza la extracción de características mediante LDA y PCA. El método *LDA* lo realizamos sobre nuestra matriz resultante de 12 variables ya que con la matriz de 16 variables no pudo ser posible por las propiedades de la matriz. En la *Ilustración 1* aparece el resultado de la división entre clases utilizando *LDA*. Sin embargo, el rendimiento de este método sobre nuestros datos hace que terminemos por considerar como mejores resultados los obtenidos con la adquisición de *PCA* en las *Ilustraciones 2 y 3.*

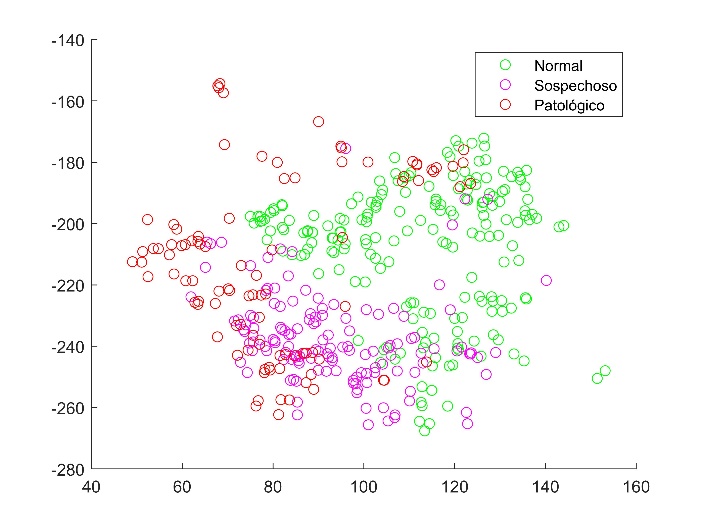
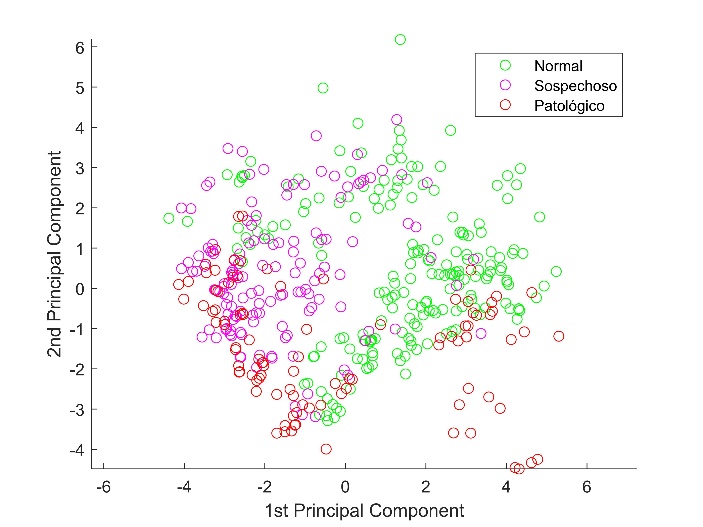
**

Ilustración 1. LDA con 12 variables.

Seguidamente se realiza *PCA* con las 16 variables y otro con las 12 variables. Para ello, partimos de los datos previamente tipificados ya que hemos obtenido mejores resultados en nuestro entrenamiento. Al representar los datos obtenidos con *PCA* mediante *Pareto*, resulta que con las 9 componentes principales de *PCA* se representa el 95% varianza. Y al proyectar los valores de las dos primeras componentes principales mediante un gráfico de dispersión podemos apreciar una ligera diferencia entre ambas ejecuciones en cuanto a la concentración de los datos. Siendo ésta mayor en el caso del *PCA* con 12 variables.



Ilusrtación 2. PCA con 16 variables.

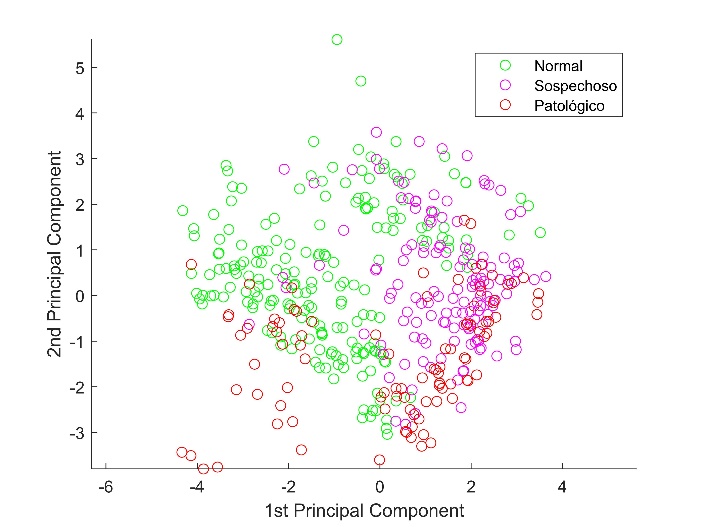


Ilustración 3. PCA con 12 variables.

Una vez que hemos terminado de analizar todas las variables procedemos a la clasificación con: LDA, QDAy K-vecinos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | LDA | QDA | K-vecinos  (K=1) | K-vecinos  (K=45) | K-vecinos  ) |
| Fscore (16) | 25.00 | - | **18.18** | 36.36 | 28.03 |
| PCA (16) | 27.58 | 27.17 | 23.84 | 31.60 | 28.89 |
| Wrappers (12) | 26.52 | - | **15.91** | 27.27 | 23.48 |
| PCA (12) | 25.71 | 24.45 | 23.10 | 26.53 | 28.89 |

*Tabla 1. Errores obtenidos con los modelos para diferentes reducciones de dimensionalidad.*

Los menores errores se obtienen para el modelo de K-vecinos=1, y con los datos habiendo aplicado los métodos directos e indirectos, *wrappers* y *fscore* respectivamente. Asimismo, con PCA se obtiene el mejor resultado con 12 variables y K-vecinos=1.

Posteriormente se entrenan estos tres modelos con K-vecinos =1 y con los métodos *HoldOut* (35%), *K-Fold* (K=15) y *Leave-One-Out* para ver la menor tasa de error.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Método HoldOut | Método K-Fold | Método LeaveOut |
| Fscore (16) | 20.61 | 17.48 | 21 |
| Wrappers (12) | 16.64 | **12.86** | **13** |
| PCA (12) | 23.62 | 20.97 | 17 |

Tabla 2. *Errores obtenidos de los modelos con los métodos HoldOut, K-Fold y LeaveOut*

El error más pequeño se da con las 12 variables obtenidas con *Wrappers* utilizando el método *K-Fold*. Sin embargo, no hay gran diferencia con el método *LeaveOut*, pero este método puede sobre ajustar nuestros datos ya que entrena con todas las observaciones salvo una.

Una vez que tengamos los datos de validación volveremos a ejecutar el modelo predictivo para ver la tasa de error resultante.

1. Discusión

El objetivo de este estudio era el desarrollo de un Sistemas Ayuda a la Decisión Médica que pudiera utilizarse para la predicción del estado fetal sirviendo como soporte al posterior diagnóstico. Para ello hemos desarrollado diversos modelos que más tarde hemos comparado para escoger aquellos que presentaran menor tasa de error y mejor ajuste. La evaluación de los modelos predictivos mostrados nos indica que la mejor aproximación y ajuste de los datos se consigue utilizando un modelo basado en memoria como es el K-vecinos, este modelo almacena las observaciones para utilizarlas como prototipos, no paramétrico por lo que no hace suposiciones explícitas sobre la forma funcional de los datos, evitando los peligros de la distribución subyacente de los datos, tiene además una precisión bastante alta ya que es insensible a los valores atípicos, aunque puede verse afectada por el ruido o las características irrelevantes.

Así, aunque el coste del aprendizaje es nulo, el coste computacional si es mucho mayor debido a la aplicación de las reglas de clasificación por vecindad que dependen de la dimensionalidad de los datos y del tamaño del conjunto de referencia, que pese a todo no nos ofrece un mecanismo para decidir automáticamente el valor óptimo de k, pues dependerá de cada conjunto de datos. Por lo que cuando tengamos los datos para validar el modelo podríamos obtener que con otro valor de K tenemos mejores resultados.

1. Conclusión

Resulta claro que tras el análisis y extracción de características de la muestra, del gran número de características presentadas, algunas al presentar una varianza 0 o muy similar a otra variable por lo que pueden ser obviadas para el entramiento del modelo siendo estas tras una primera aproximación indirecta *AC (número de aceleraciones por segundo)*, *DL (número de desaceleraciones de la luz por segundo)*, *DS (número de deceleraciones severas por segundo)*, *DP (número de deceleraciones prolongadas por segundo)*  y *Nzeros (número de ceros del histograma)*, y a las que añadimos también *MSTV (valor medio de la variabilidad a corto plazo )*, *Width (anchura del histograma de FHR)*, *Min (mínimo de histograma de la FCF)*, *Median (mediana del histograma)* tras una aproximación directa. A parte de estas aproximaciones también realizamos una extracción de variables con PCA.

Más adelante para la clasificación con LDA, QDA y K-vecinos, obtuvimos que la menor tasa de error y el major ajuste se conseguía con K-vecinos cuando K=1, además de que la extracción de características con PCA era más optima cuando solo contábamos con 12 variables.

Finalmente a falta de los datos de validación, concluimos que el entramiento se puede llevar a cabo con el método HoldOut, método K-Fold, o el método LeaveOut, obteniendo mejor resultados para el método K-Fold

1. Bibliografía

[1] D. Lobach, A. Mckibbon, and M. C. Gibbons, “National Web-Based Teleconference on Findings from Evidence-Based Practice Centers for Health IT,” 2011.

[2] R. M. Grivell, Z. Alfirevic, G. M. Gyte, and D. Devane, “Antenatal cardiotocography for fetal assessment,” Cochrane Database Syst. Rev., Sep. 2015.

[3] C. Sáez, “Unidad 3.3. Introducción a los CDSS tipo IV y análisis exploratorio.” 2019.

[4] C. Sáez, “Sistemas de Ayuda a la Decisión Basados en Datos Biomédicos: Métodos de Aprendizaje,” 2019.

[5] R. DE Teledetección, “55 Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) Análisis de métodos de validación cruzada para la obtención robusta de parámetros biofísicos Analysis of cross-validation methods for robust retrieval of biophysical parameters,” vol. 44, pp. 55–65, 2015.

[6] C. Sáez, “Estrategias y métricas de evaluación de modelos predictivos y SADM.” 2019.