Introducción a la minería de datos en banca y finanzas.
Documento de apoyo.

Ricardo Restrepo Lopez [*] Jose Antonio Solano [**]

Juan Fernando Rendón [***]

2 de octubre de 2014

Resumen

El reconocimiento automático de patrones complejos y el diseño de algoritmos 'inteligentes' de toma de decisiones basados en datos empíricos ha revolucionado todas las áreas de aplicación tecnológicas en los últimos años. Hemos presenciado el paso de la era algorítmica basada en 'reglas' a la era basada en 'aprendizaje de máquina'. Por supuesto, esta revolución habría de incorporarse rápidamente a la industria financiera: el trading automatizado, el mercado de señales y el trading de alta frecuencia son ejemplos visibles de esta revolución.

Esta capacitación permitirá a los participantes conocer los múltiples beneficios que ofrece la minería de datos y el aprendizaje de máquina en los sectores financiero y bancario.

En particular, conocerán cómo la minería de datos resulta indispensable en la actualidad para poder anticipar patrones y comportamientos de clientes, del mercado o de los competidores.

CÁMARA DE BANCOS E INSTITUCIONES FINANCIERAS DE COSTA RICA

ACADEMIA BANCARIA CENTROAMERICANA

[*] ricardo.restrepo@mathdecision.com
[**] jose.solano@mathdecision.com
[***] juan.rendon@mathdecision.com

Índice

1. Programa de la capacitación

1.1. Día 1

1. Introducción: Minería de datos, modelos estadísticos, aprendizaje 'supervisado' y 'no supervisado', concepto de error, cross-validation. Modelos lineales.

2. Regularización: *Ridge*, *Lasso* y *Elastic Net*. Influencia de índices financieros.

3. Redes neuronales: Perceptron. Árboles de decisión. Mercadeo dirigido. Aprobación de crédito. Series de tiempo financieras.

1.2. Dia 2

1. Máquinas de soporte vectorial. Selección de portafolios. Predicción en mercados cambiarios. Series de tiempo financieras.

2. Métodos de clustering: K-means, affinity propagation, clustering jerárquico. Clasificación de condiciones del mercado. Concentración en carteras.

3. Detección de novedades y anomalías. Fraude en transacciones financieras.

2. Índice de ejemplos

En el desarrollo de esta capacitación usaremos las siguientes bases de datos de ejemplo disponibles públicamente.

2.1. Mercadeo de producto bancario

Los datos están relacionados con campañas de marketing directo (llamadas telefónicas) de una institución bancaria portuguesa. El objetivo es la clasificación para predecir si el cliente va a suscribir un depósito a plazo (variable y).

`https://archive.ics.uci.edu/ml/datasets/Bank+Marketing`

2.2. Indice de bolsa

Comparacion del indice de la bolsa de estanbul frente a indices varios SP, DAX, FTSE, NIKKEI, BOVESPA, MSCEEU, MSCIEM.

`https://archive.ics.uci.edu/ml/datasets/ISTANBUL+STOCK+EXCHANGE`

2.3. Base de datos de *German Credit Data*

Esta base de datos clasifica la población a través de una serie de atributos como sujetos de crédito buenos o malos.

`https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)`

2.4. Base de datos de *Australian Credit Approval*

Base de datos de apliaciones a tarjetas de crédito.

`https://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval)`

2.5. Advances and Declines

Datos historicos de cierre de bolsa

`http://unicorn.us.com/advdec/`

2.6. Ivolatility

Precios historicos de opciones y volatilidad implicada.

`http://www.ivolatility.com/`

2.7. OECD.Stat Extracts

Datos y metadatos de estadisticas sociales y economicas de paises OCED

`http://stats.oecd.org/index.aspx`

2.8. Google finance

Precios historicos recientes de cierre de bolsa

`https://www.google.com/finance`

3. Introducción

Recoger y mantener grandes colecciones de datos es una ardua tarea, pero extraer información útil de tales colecciones es un reto aún más complejo. El fenómeno del Big Data no solo requiere cambios en las herramientas que usamos para el análisis predictivo, sino que además requiere cambios en nuestra manera de pensar el problema de la extracción de conocimiento de los datos.

Tradicionalmente, este 'tratamiento de datos' ha estado dominado por diversos métodos estadísticos de error y ensayo, pero tal metodología se torna imposible cuando las colecciones de datos son grandes y heterogéneas, ya que estas se enfocan en análisis estáticos limitados a muestras que están fijas en el tiempo y cuyos resultados frecuentemente pierden validez rápidamente y dejan de ser confiables. Sin embargo, ingeniosas alternativas han sido propuestas recientemente por un campo de investigación de rápida expansión: el aprendizaje de máquina (machine learning) y el tratamiento analítico de datos (data analytics), los cuales se condensan bajo el término de minería de datos (data mining).

Esta área, la cual combina novedosas teorías provenientes de la estadística y las ciencias de la computación en aplicaciones emergentes de la industria, se enfoca en el desarrollo de algoritmos rápidos y eficaces para el procesamiento de datos en tiempo real con el propósito principal de entregar predicciones confiables que sean usables para la empresa. Para nombrar algunas de estas aplicaciones, pensemos en industrias que requieren servicios tales como: recomendación de productos a usuarios, segmentación de clientes, detección de fraude o prevención de abandono de clientes.

Las técnicas de minería de datos proveen soluciones a tales aplicaciones a través de métodos genéricos que difieren de las soluciones estadísticas tradicionales. Su énfasis en tiempo real y escalamiento, usando métodos completamente automatizados que simplifican enormemente el análisis de datos tradicional han hecho que esta área halla penetrado fuertemente en la industria en años recientes y sea un nuevo factor de competencia entre los actores del mercado.

3.1. ¿Qué es minería de datos?

La minería de datos es un conjunto de algoritmos y técnicas diseñadas con el propósito de inferir reglas y patrones no triviales a partir de grandes volúmenes de datos.

La minería de datos se utiliza principalmente para tareas de clasificación (predicción de variables discretas) , regresión (predicción de variables continuas), estimación de densidad de distribución y clustering (técnicas de agrupamiento de datos).

Para aplicar las técnicas de minería de datos, se requieren de grandes cantidades de datos que permitan encontrar patrones difícles de intuir o hallar con otro tipo de métodos tradicionales.

3.2. Modelos paramétricos vs no paramétricos

En términos generales, un modelo es un esquema teórico, generalmente expresado en forma matemática, de un sistema o de una realidad compleja, (la

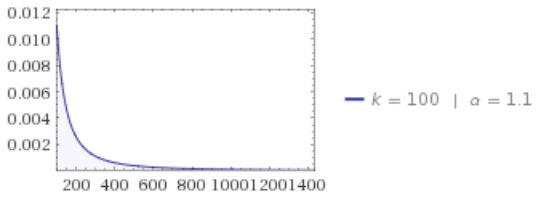
evolución económica de un país, el comportamiento de un cliente, etc) que se elabora para facilitar su comprensión y estudiar su comportamiento.

Un modelo estadístico, en particular, permite agregar dos aspectos claves en el modelamiento: *aleatoriedad* e *incertidumbre*.

Por ejemplo, la magnitud de las pérdidas ocasionadas por clientes que caen en *default* se suele modelar con una distribución de probabilidad tipo Pareto [8]. En este caso, el monto de las pérdidas dependerá de la forma de la distribución, en particular su densidad de probabilidad estará dada por la función

$$f(x) = \frac{\alpha m^{\alpha}}{x^{\alpha+1}},$$

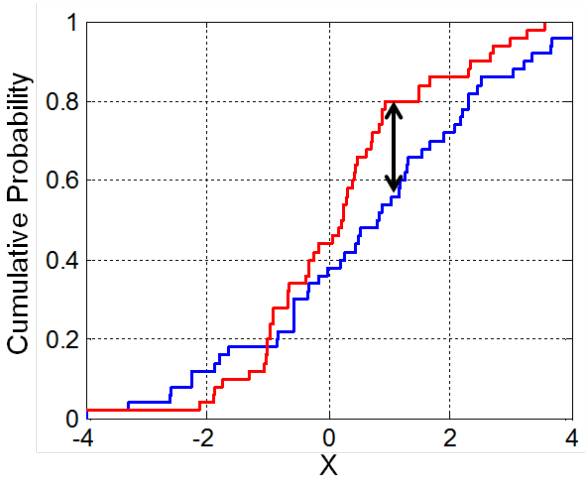
la cual depende de los parámetros m (escala) y α (forma).



Tal tipo de modelos se denominan paramétricos pues corresponden a una descripción predeterminada (una familia parametrizada de distribuciones), a la que el modelo debe obedecer.

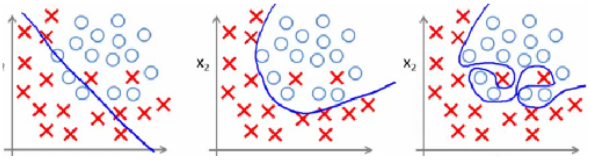
Sin embargo, una manera alternativa de modelar el fenómeno anterior consiste en elaborar un pronóstico de acuerdo a la experiencia histórica de la institución sin tratar de ajustar los datos presentados a una función de distribución predefinida.

Cuando los datos tienen una clasificación numérica pero ninguna interpretación clara, tales como la asignación de preferencias, puede hacerse necesario el uso de métodos no paramétricos. Como los métodos no paramétricos hacen menos suposiciones, su aplicabilidad es mucha más amplia que la de los métodos paramétricos correspondientes. En particular, se pueden aplicar en situaciones en las que el conocimiento de la aplicación en cuestión es reducido (por tener pocoso datos o experiencia). Una ventaja adicional de los métodos paramétricos es que, debido a la dependencia de un número menor de hipótesis, presentan una mayor robustez (baja sensibilidad a errores de estimación).



3.3. Error y validación de modelos

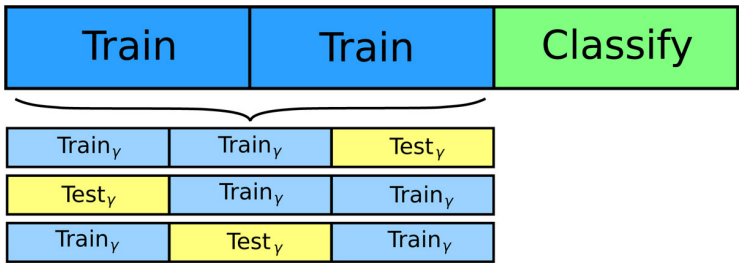
En minería de datos, *overfitting* ocurre cuando un modelo estadístico describe el error aleatorio o ruido en lugar de la relación subyacente. El *overfitting* ocurre generalmente cuando el modelo es excesivamente complejo, tal es el caso cuando se tienen demasiados parámetros relativos al número de observaciones. Un modelo que ha sido sobreajustado generalmente tendrá pobre rendimiento predictivo, ya que puede exagerar fluctuaciones menores en los datos.



El fenómeno *underfitting* (en el extremo izquierdo) ocurre cuando un estimador no es lo suficientemente flexible como para captar las tendencias subyacentes en los datos observados. Contrario al *overfitting* (en el extremo derecho) que captura tendencias ilusorias en los datos.

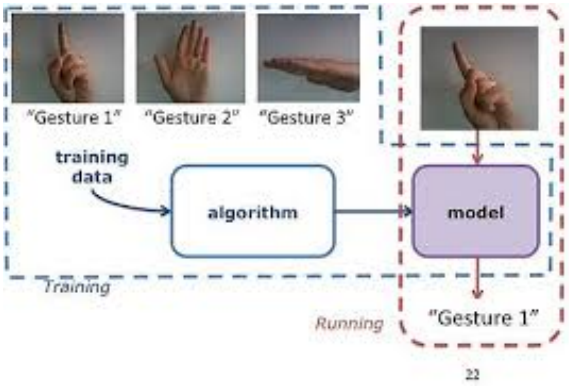
Otro aspecto de gran importancia para la evaluación y selección del modelo es la validación cruzada.

Cuando se ha construido un modelo es necesario evaluar la capacidad de éste para producir resultados acordes con la realidad del fenómeno. Este proceso se lleva ha cabo separando los datos en conjunto de datos de entrenamiento y prueba. El conjunto de datos de entrenamiento se utiliza para ajustar el modelo y el conjunto de datos de prueba para confirmar la precisión del modelo.



4. Métodos de aprendizaje supervisado

En minería de datos, los métodos de *aprendizaje supervisado* consisten en una variedad de modelos y algoritmos que tienen el propósito de 'aprender' o 'inferir' relaciones no triviales de los datos. De manera particular, tales modelos buscan relacionar una serie de variables de *entrada* (variables cuya información estaría disponible) con una o varias variables de salida (variables cuya información se desea inferir). Tal relación se expresa en la forma de un modelo de inferencia (función lineal, red neuronal, esquema de clasificación, etc). El modelo se ajusta (o 'aprende') de una serie de datos *etiquetados*, esto es, para los que se conoce el valor de la variable de interés (usualmente datos históricos) y el modelo ajustado se emplea luego para inferir la variable o característica de interés para datos no etiquetados o datos futuros (para los que no se conoce el valor de la variable de interés, pero se conocen las variables de entrada).



5. Algoritmos de regresion

5.1. Regresion lineal simple

Un modelo de *regresion lineal simple* trata de inferir una función que relaciona la variable objetivo con las variables de entrada mediante la relación

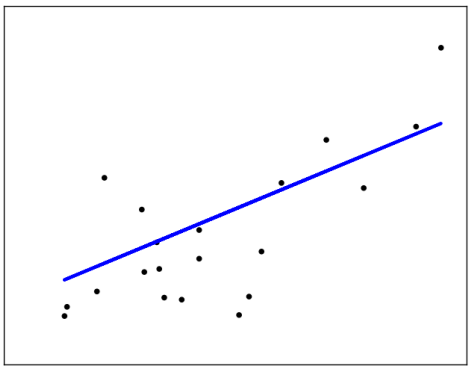
$$\hat{y} = \sum_{i=1}^{d} \mathbf{x}_i \beta_i + \beta_0 + \epsilon,$$

esto es, mediante una relación lineal entre las variables de entrada y la salida. En este modelo, β es el vector de coeficientes, ϵ es la variable aleatoria que modela

el error de la regresión, $\hat{y}$ es el valor estimado de la variable de interés y x es el valor censado de las variables de entrada.

Uno de los modelos de mayor difusión y uso para establecer relaciones entre variables de entrada y variables de salida, es el modelo de regresión lineal simple, dado que parte del supuesto de que las variables de entrada y de salida tienen una relación proporcional directa.

El análisis del modelo de regresión lineal simple se toma como punto de partida, ya que provee conceptos básicos necesarios para el entendimiento de modelos que plantean relaciones mas complejas.



Para ilustrar su funcionamiento trabajaremos sobre los ejemplo que se encuentran en la carpeta regresion-simple

5.2. Regularización

Las estimaciones de los coeficientes por mínimos cuadrados ordinarios se basan en la independencia de los términos del modelo. Cuando los términos se correlacionan y las columnas de la matriz de diseño X tienen una dependencia aproximadamente lineal, puede ocurrir que la matriz de diseño se vuelva singular y como resultado, la estimación por mínimos cuadrados se vuelve demasiado sensible a los errores aleatorios en la respuesta observada, produciendo una gran varianza. Esta situación de multicolinealidad puede surgir, por ejemplo, cuando los datos se recolectan sin un diseño experimental.

La regresión *Ridge* aborda algunos de los problemas de mínimos cuadrados ordinarios mediante la imposición de una penalidad en el tamaño de los coeficientes. Los coeficientes *Ridge* minimizan la suma de cuadrados de los residuos penalizada,

$$\min_{w} ||Xw - y||_2{}^2 + \alpha||w||_2{}^2$$

Aquí, $\alpha \geq 0$ es un parámetro que controla la cantidad de contracción: cuanto mayor es el valor de α, mayor es la cantidad de contracción y por lo tanto los coeficientes se hacen más robustos ante colinealidad.

Para ilustrar su funcionamiento trabajaremos sobre los ejemplo que se encuentran en la carpeta regresion-ridge

La regresión *Lasso* es un modelo lineal que estima los coeficientes dispersos. Es útil en algunos contextos debido a su tendencia a preferir soluciones con menos valores de los parámetros, reduciendo efectivamente el número de variables en los que la solución dada es dependiente. Por esta razón, el modelo *lasso* y sus variantes son fundamentales para el campo de la adquisición compresiva (*Compressed Sensing*). Bajo ciertas condiciones, se puede recuperar el conjunto exacto de pesos distintos de cero.

Matematicamente, se compone de un modelo lineal cuya función objetivo a minimizar es:

$$\min_{w} \frac{1}{2n_{samples}} ||Xw - y||_2^2 + \alpha ||w||_1$$

Para ilustrar su funcionamiento trabajaremos sobre los ejemplo que se encuentran en la carpeta regresion-lasso

ElasticNet es un modelo de regresión lineal que combina las ventajas de la regresión *ridge* y *lasso*. Esta combinación permite el aprendizaje de un modelo disperso, donde pocos de los pesos son diferentes cero como *Lasso*, al tiempo que se mantienen las propiedades de regularización de *Ridge*.

Elástico-net es útil cuando hay varias características que se correlacionan entre sí. Con *Lasso* es probable elegir uno de ellos al azar, mientras que con *elástica-net* es probable elegir ambos. La función objetivo a minimizar en este caso es

$$\min_{w} \frac{1}{2n_{samples}} ||Xw - y||_2^2 + \alpha\rho ||w||_1 + \frac{\alpha(1-\rho)}{2} ||w||_2^2$$

Para ilustrar su funcionamiento trabajaremos sobre los ejemplo que se encuentran en la carpeta regresion-eslastic

6. Algoritmos de clasificación

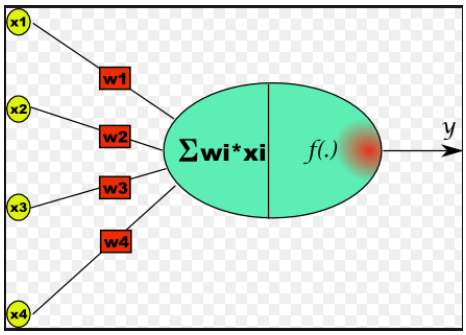
6.1. Redes neuronales

En el área de aprendizaje de máquina, los modelos de redes neuronales artificiales, son modelos computacionales inspirados en la dinámica de funcionamiento del sistema nervioso central animal. Tales modelos, bajo los supuestos adecuados, son adecuados para implementar aprendizaje de máquina y predicción de patrones. De manera general, una red artificial neuronal es presentada como un sistema de 'neuronas' interconectadas las cuales pueden comutar valores ante una entrada determinada y ajustar sus propios parámetros de cómputo.

Las redes neuronales artificiales (RNA) procesan información con características de desempeño similares a las del cerebro humano, en particular a las redes neuronales biológicas. En este orden de ideas se puede decir que las RNA son modelos de las redes neuronales biológicas.

Las RNA se desarrollaron como una generalización de los modelos matemáticos basados en los siguientes aspectos:

- El procesamiento de información ocurre en elementos llamados neuronas.
- Las neuronas se transmiten información a través de conexiones.
- En cada neurona se aplica una función de activación a las entradas.



El primer modelo de red neuronal fue propuesto en 1943 por McCulloch y Pitts [11] en términos de un modelo computacional de actividad nerviosa. El modelo citado consistía en un modelo binario donde cada neurona tiene un umbral prefijado. Desarrollos posteriores de Jhon Von Neumann, Marvin Minsky, Frank Rosenblatt, Kohonen, y otros, tomaron como ejemplo el modelo de McCulloch y Pitts, dando origen al área actualmente denominada redes neuronales artificiales (RNA).

Las RNA se utilizan para extraer patrones y detectar relaciones que son difíciles de apreciar por humanos u otras técnicas computacionales. Una RNA entrenada puede usarse como un experto para categorizar la información que se ha dado para su análisis. Este experto puede usarse para proveer pronósticos de nuevas situaciones.

Algunas de las aplicaciones usuales de las RNA en el sector financiero son la evaluación de la asignación de crédito, identificación de firmas, lectores de documentos, valuación de bonos corporativos, predicción de tipo de cambio, análisis de uso de la línea de crédito, entre otras.

6.1.1. Red neuronal Perceptron

El modelo básico de un esquema de clasificación consiste en inferir una función f de tal manera que $y = f(z)$ denote la salida o grupo de clasificación para un vector de entrada z. El algoritmo perceptron, en particular, procura hallar una función de la forma

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

donde w es el *vector de pesos* de cada variable y b es el término de *sesgo*.

Un problema típico consiste de un conjunto $D = \{(x_1, d_1), \ldots, (x_s, d_s)\}$ de *muestras de entrenamiento* donde cada término x_j corresponde a un registro de datos de entrada (que puede contener varios campos) y d_j es la salida requerida.

Atributo 1	Atributo 2	Atributo 3	Aprobacion de credito
25,67	3,25	2,29	0
31,67	16,165	3	1
30,08	1,04	0,5	0
21,92	0,54	0,04	1
64,08	20	17,5	1
42,75	4,085	0,04	0
34,17	9,17	4,5	1
27,25	1,665	5,085	1
24,75	13,665	1,5	0
42,75	3	1	0
35,75	2,415	0,125	0

El algoritmo perceptron es un algoritmo de clasificación *en linea*, esto es, explora una a una las muestras de entrenamiento y ajusta el modelo de forma progresiva. De manera más precisa, el algoritmo perceptron opera de las siguiente forma:

- Inicializa los pesos y el término de sesgo (por lo general, en 0)
- Se define una tasa de aprendizaje $\alpha \in (0, 1]$
- En el paso j del proceso se calcula $y = f(x_j)$ con el modelo de clasificación estimado hasta el momento.
- El peso de la variable i se ajusta de acuerdo al siguiente esquema:

$$w(i) \leftarrow w(i) + \alpha x_j(i)(d_j - y_j)$$

- El sesgo se ajusta de la siguiente manera:

$$b \leftarrow b + \alpha(d_j - y_j)$$

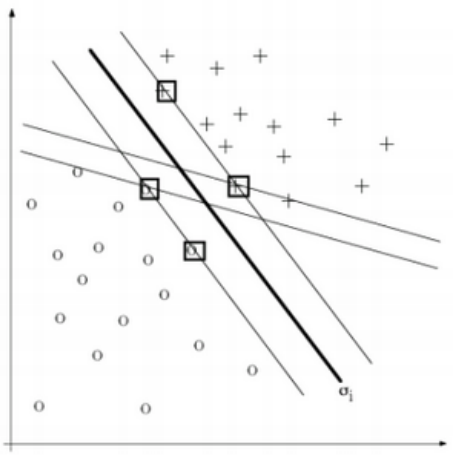
- Se repite hasta obtener convergencia

Para ilustrar su funcionamiento trabajaremos sobre los ejemplo que se encuentran en la carpeta perceptron

6.2. Maquinas de soporte vectorial

Este es un algoritmo de clasificación cuyo propósito es separar las clases por medio de un hiperplano que maximiza la distancia del punto más cercano de cualquier clase. Llamamos a esta distancia el *margen*. Los puntos en el margen son llamados *vectores de soporte*.

Suponga que las clases son linealmente separables como en la figura.



El hiperplano que maximiza el margen se encuentra resolviendo el problema

$$\arg \min_{(\mathbf{w},b)} \frac{1}{2}\|\mathbf{w}\|^2$$

sujeto a las restricciones

$$y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1.$$

En este caso, los planos óptimos separadores serán los planos $w \cdot x - b = 1$ y $w \cdot x - b = -1$.

El problema resultante es un problema de programación cuadrática (función objetivo cuadrática con restricciones lineales), un tipo estándar de problema de optimización para el cual existen algoritmos de rápido computo que hallan el óptimo global.

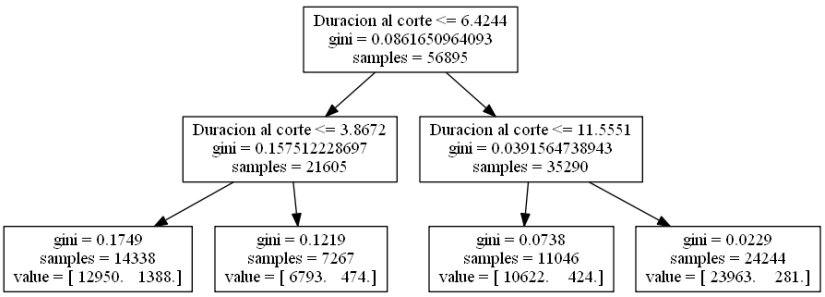
Este método pertenece a la categoría de métodos de clasificación llamados de 'Kernel', los cuales han revolucionado la minería de datos desde su introducción en 1992 por Vapnik y colaboradores [1]. En particular, este método ha encontrado uso en aplicaciones tales como

1. Pronóstico de series de tiempo financieras [7, 15].
2. Modelos de quiebra [12, 14].
3. Seleccion de portafolios [3, 6].
4. *Scoring* de crédito [5, 10].

Para ilustrar su funcionamiento trabajaremos sobre los ejemplo que se encuentran en la carpeta svm

6.3. Árboles de decisión

Los árboles de decisión consisten en un método *no-paramétrico* de aprendizaje supervisado el cual es usado para clasificación y regresión. El objetivo es crear un modelo que predice el valor de una variable objetivo por medio de reglas simples de decisión inferidas a partir de las variables de entrada.



Algunas ventajas de los árboles de decisión son las siguientes:

- Son simples de entender e interpretar, en particular estos se pueden visualizar de forma sencilla.
- Requieren poca preparación de los datos. Otras tecnicas a menudo requieren normalización de los datos, creación de variables dummy (categoricas) y eliminación de espacios en blanco.

- Tienen la capacidad de manipular tanto datos numéricos como datos categóricos. Otro tipo de técnicas normalmente se especializan en analizar conjuntos de datos que tienen solo un tipo de variable.
- Soporta problemas de clasificación múltiple.
- Utiliza modelos de caja blanca. Si una situación dada es observable en el modelo, la explicación para la condición es fácilmente explicada por el árbol de decisión. En contraste con los modelos de caja negra (e.g redes neuronales artificiales), en donde los resultados pueden ser mas difíciles de interpretar.
- Se pueden crear arboles de decisión excesivamente complejos que no generalizan bien los datos. A esto se le llama sobreajuste.
- La optimización de arboles de decisión es compleja incluso para conceptos sencillos. Por esto, la mayoría de algoritmos que hallan árboles de decisión se basan en esquemas heurísticos que no garantizan convergencia.
- En algunas ocasiones el árbol de decisión ajustado tiene una complejidad innecesaria. Esto es, ciertas relaciones sencillas son complejas de representar en un árbol de decisión.
- El entrenamiento de arboles de decisión crea sesgos si alguna de las clases es dominante, por esto se recomienda balancear el conjunto de datos previamente antes de hacer el ajuste del arbol de decisión.

Para ilustrar su funcionamiento trabajaremos sobre los ejemplo que se encuentran en la carpeta arboles

7. Metodos de aprendizaje no supervisado

7.1. Clustering

El objetivo de un problema de separación en Clusters es descubrir grupos de ejemplos similares en los datos, los cuales se condensan en una entidad de caracteristicas similares llamada *Cluster*. La idea es que los objetos que se encuentran en el mismo grupo (Cluster) son mas similares (en algúna caracteristica) que aquellos que están en otros grupos.

La noción de 'cluster', en realidad no puede ser definida de manera precisa. Por esto mismo, existe una gran variedad de algoritmos de clustering, cada uno adaptado a la noción adecuada del concepto de cluster.

7.2. Método de K-means

El algoritmo de clusters de datos de *K Means* trata de separar un conjunto de datos en n grupos con dispersión similar entre los datos (igual varianza). En particular, este algoritmo busca minimizar la dispersión existente dentro de cada uno de los grupos. La medias de cada clusters es llamada 'centroides' (notese que el centro no es necesariamente un dato del conjunto). De manera precisa,

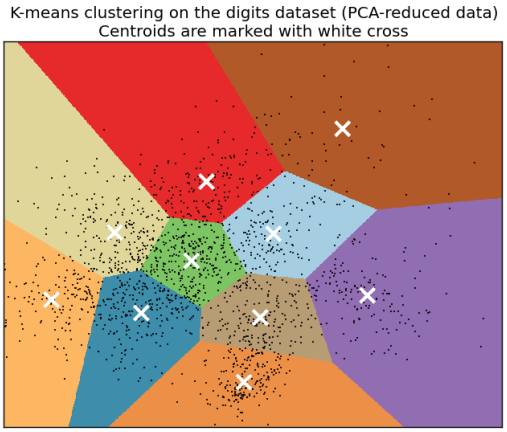


Figura 1: Clusters por K-means

el algoritmo de K means permite seleccionar los centroides que minimizan la dispersión dentro de los clusters, esto es, trata de minimizar:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_j - \mu_i||^2)$$

El algoritmo de K-means es comunmente usado en concentración de carteras, discriminación de clientes y busqueda de anomalias. Otro uso común es el de clasificación de transacciones a partir del libro de ordenes.

El algoritmo de Lloid [9] es un esquema iterativo para hallar los centroides que minimizan la dispersión. Este algoritmo opera de la siguiente manera:

- Se inicializan los centroides $\mu_1, \ldots, \mu_k$.
- Se hallan los clusters $C_1, \ldots, C_k$ para estos centroides. En el cluster C_i estarán los datos x cuya distancia al centroide μ_i es menor que la distancia a cualquier otro centroide μ_j.
- Se actualizan los centroides de tal manera que

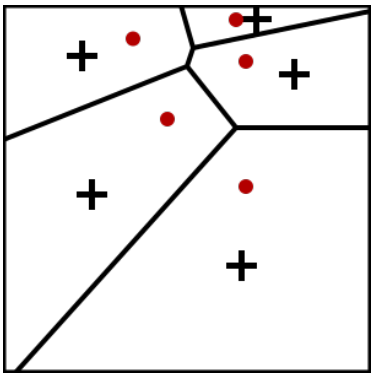
  $$\mu_i \leftarrow \mu(C_i),$$

  donde $\mu(C_i)$ es la media de los puntos en el cluster C_i.
- Se repite el proceso hasta convergencia

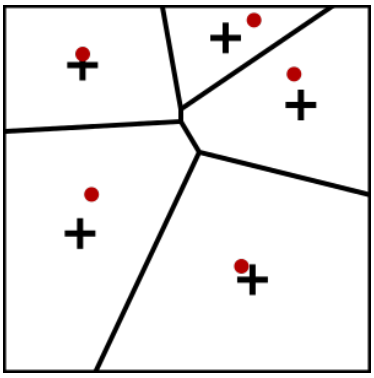
Para ilustrar su funcionamiento trabajaremos con el ejemplo ejemplo `kmeans univariado excel.xlsx` en la carpeta kmeans

7.3. Affinity propagation

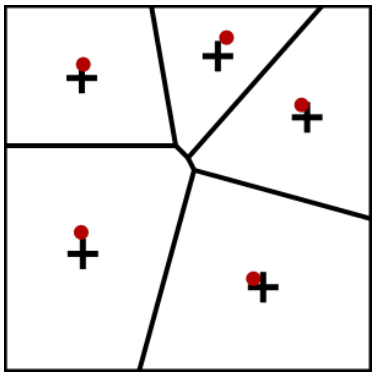
El algoritmo de clusters de datos *AffinityPropagation* crea clusters a través de un algoritmo de transferencia de mensajes entre pares de muestras. Un conjunto



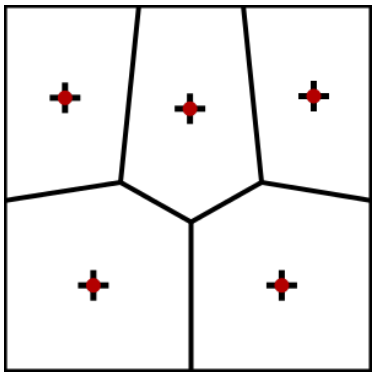
(a) Inicialización de los centroides



(b) Primera iteración



(c) Segunda iteración



(d) Tercera iteración

de datos es entonces descrito utilizando un número pequeño de ejemplares que se identifican como los más representativos de otras muestras. Los mensajes enviados entre pares representan la idoneidad de que una muestra sea el ejemplar de la otra, que se actualiza en respuesta a los valores de otros pares. Esta actualización ocurre iterativamente hasta la convergencia, momento en el que se eligen los ejemplares finales, y por lo tanto se da el agrupamiento-clustering final.

El algoritmo opera de la siguiente manera: Los mensajes enviados entre los puntos pertenecen a una de dos categorías. La primera es la responsabilidad $r(i, k)$, que es la evidencia acumulada de que la muestra k debe ser el ejemplar para la muestra i. El segundo es la disponibilidad $a(i, k)$, que es la evidencia acumulada de que la muestra i deba elegir a la muestra k como su ejemplar, considerando la decisión de las otras muestras. De esta manera, un ejemplar es elegidos por las muestras si (1) es lo suficientemente 'similar' a muchas muestras y (2) es elegido por muchas muestras para ser representante de ellas.

Más formalmente, los mensajes de responsabilidad y disponibilidad se actualizan de la siguiente forma:

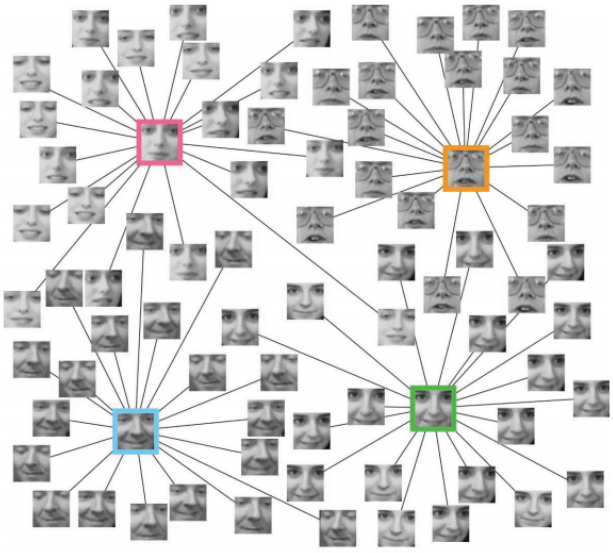


Figura 2: Ejemplares resultantes de *Affinity propagation*

AFFINITY PROPAGATION

INPUT: $\{s(i,j)\}_{i,j\in\{1,\ldots,N\}}$ (data similarities and preferences)

INITIALIZE: set 'availabilities' to zero *i.e.* $\forall i,k\colon a(i,k)=0$

REPEAT: responsibility and availability updates until convergence

$$\forall i,k\colon\ r(i,k)=s(i,k)-\max_{k':k'\neq k}\left[s(i,k')+a(i,k')\right]$$

$$\forall i,k\colon\ a(i,k)=\begin{cases}\sum_{i':i'\neq i}\max[0,\ r(i',k)],\ \text{for } k=i\\ \min\left[0,\ r(k,k)+\sum_{i':i'\notin\{i,k\}}\max[0,\ r(i',k)]\right],\ \text{for } k\neq i\end{cases}\qquad(3.15)$$

OUTPUT: cluster assignments $\hat{\mathbf{c}}=(\hat{c}_1,\ldots,\hat{c}_N)$, $\hat{c}_i=\text{argmax}_k\left[a(i,k)+r(i,k)\right]$

Note: $\hat{c}$ may violate $\{f_k\}$ constraints, so initialize k-medoids with $\hat{c}$ and run to convergence for a coherent solution.

Para ilustrar su funcionamiento trabajaremos sobre los ejemplos que se encuentran en la carpeta affinity

7.4. Detección de novedades y anomalias

Consideremos un conjunto datos dentro de un conjunto de n observaciones de la misma distribución descrita por p características. Si añadimos una observación más a ese conjunto de datos. ¿Es la nueva observación tan diferente de los otros que podemos dudar que sigue la misma distribución? O por el contrario, es tan similar a las otras que no podemos distinguirla de las observaciones originales? Esta es la cuestión abordada por las herramientas y métodos de detección de la novedades.

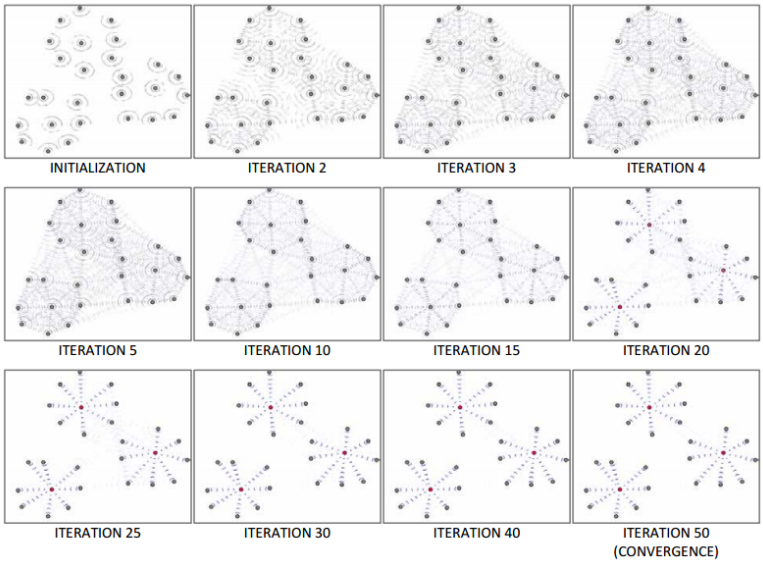
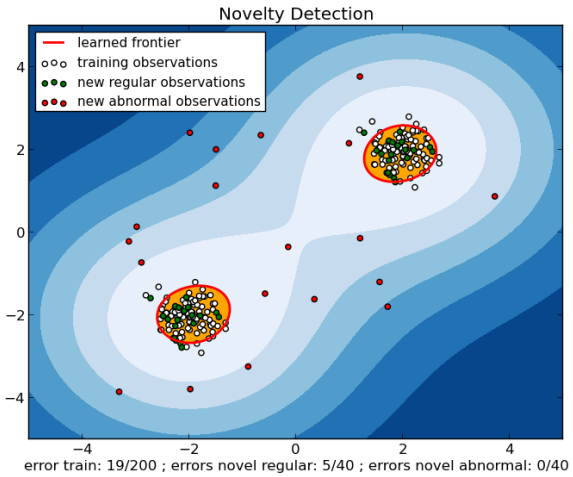


Figura 3: Algoritmo de *Affinity propagation*

En general, se trata de aprender, cerca de la frontera que delimita el contorno de la distribución de las observaciones iniciales, embebidas en el espacio p-dimensional. Entonces, si nuevas observaciones caen dentro del subespacio delimitado por la frontera, se consideran como procedentes de la misma población que las observaciones iniciales. De lo contrario, si se caen fuera de la frontera, se puede decir que son anormales con una confianza dada en nuestra evaluación.

El método SVM de una clase se emplea para tal fin. En la carpeta novedades encontraremos ejemplos que ilustran su uso.

El hallazgo de datos anomalos (es decir, datos posiblemente erroneos que deben ser descartados de la muestra), sigue la misma metodologia.



Referencias

[1] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. .A training algorithm for optimal margin classifiers."Proceedings of the fifth annual workshop on Computational learning theory. ACM, 1992.

[2] Cao, Lijuan, and Francis EH Tay. "Financial forecasting using support vector machines."Neural Computing & Applications 10.2 (2001): 184-192.

[3] Fan, Alan, and Marimuthu Palaniswami. "Stock selection using support vector machines."Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on. Vol. 3. IEEE, 2001.

[4] Frey, Brendan J., and Delbert Dueck. Çlustering by passing messages between data points."science 315.5814 (2007): 972-976.

[5] Huang, Cheng-Lung, Mu-Chen Chen, and Chieh-Jen Wang. Çredit scoring with a data mining approach based on support vector machines..Expert Systems with Applications 33.4 (2007): 847-856.

[6] Huang, Chien-Feng. .A hybrid stock selection model using genetic algorithms and support vector regression..Applied Soft Computing 12.2 (2012): 807-818.

[7] Kim, Kyoung-jae. "Financial time series forecasting using support vector machines."Neurocomputing 55.1 (2003): 307-319.

[8] Lindskog, Filip, and Alexander J. McNeil. Çommon Poisson shock models: applications to insurance and credit risk modelling..Astin Bulletin 33.2 (2003): 209-238.

[9] Lloyd, Stuart. "Least squares quantization in PCM.İnformation Theory, IEEE Transactions on 28.2 (1982): 129-137.

[10] Martens, David, et al. Çomprehensible credit scoring models using rule extraction from support vector machines..European journal of operational research 183.3 (2007): 1466-1476.

[11] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.

[12] Min, Jae H., and Young-Chan Lee. "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters..Expert systems with applications 28.4 (2005): 603-614.

[13] Moro, Sérgio, Paulo Cortez, and Paulo Rita. .A data-driven approach to predict the success of bank telemarketing."Decision Support Systems 62 (2014): 22-31.

[14] Shin, Kyung-Shik, Taik Soo Lee, and Hyun-jung Kim. .An application of support vector machines in bankruptcy prediction model..Expert Systems with Applications 28.1 (2005): 127-135.

[15] Van Gestel, Tony, et al. "Financial time series prediction using least squares support vector machines within the evidence framework."Neural Networks, IEEE Transactions on 12.4 (2001): 809-821.