Autor: Cuesta Alario David

Matriz de confusión

$$Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN} \qquad F-Measure = \frac{2 * Precision * Recall}{Precision + Recall} \qquad Accuracy = \frac{TP+TN}{TP+FN+FP+TN}$$

		Clase Estimada	
		Positiva	Negativa
Clase Real	Positiva	TP **A**	FN **B**
	Negativa	FP **C**	TN **D**

$$P(Positiva) = \frac{TP+FN}{TP+FN+FP+TN} + \frac{TP+FP}{TP+FN+FP+TN} \qquad Kapa = \frac{Accuracy - P(acuerdo)}{1 - Accuracy\ M2}$$

$$P(Positiva) = \frac{TN+FN}{TP+FN+FP+TN} + \frac{TN+FP}{TP+FN+FP+TN}$$

$$P(acuerdo) = P(Positiva) + P(Negativa)$$

Las instancias se evalúan individualmente y se compara la predicción con la clase real

- **A TP** → Verdaderos Positivos → instancias que eran de la clase positiva y se predijeron como positivas
- **D TN** → Verdaderos Negativos → instancias que eran de la clase negativa y se predijeron como negativo
- **B FN** → Falsos Negativos → instancias que eran de la clase positiva pero se predijeron como negativas
- **B FP** → Falsos Positivos → instancias que eran de la clase negativa pero se predijeron como positivas
- **Kapa** Mide desacuerdo entre dos estimadores $Kapa = 0$ Si no hay acuerdo entre los clasificadores que no sea lo que se esperaría por casualidad

Análisis de ROC

Permite caracterizar la calidad de los clasificadores en base a su rendimiento

Es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación

Proporciona herramientas para seleccionar los modelos posiblemente óptimos y descartar modelos subóptimos independientemente del coste de la distribución

En el punto (0 , 0) se predice todo clase negativa En el punto (1 , 1) se predice todo clase positiva En el punto (0 , 1) Se situaría el mejor método de predicción

- Si TPR = FPR obtenemos un clasificador aleatorio en el que los atributos no aportan información sobre la clase
- Podemos obtener un resultado peor que el del clasificador aleatorio si estamos por debajo de la línea TPR = FPR. En este caso la predicción del clasificador guarda cierta correlación con la realidad, pero la correlación es negativa. La solución es tomar la decisión contraria obteniendo un resultado simétrico sobre la diagonal del espacio ROC
- El mejor clasificador será aquel que tenga mayor área bajo la curva ROC

Razón de Falsos Positivos ⟶ eje X

$$FPR = \frac{FP}{FP+TN} \longrightarrow (1 - Specificity)$$

Razón de Verdaderos Positivos ⟶ eje Y

$$TPR = \frac{TP}{TP+FN} \longrightarrow Sensitivity$$

Para calcular el área de la curva de ROC lo descomponemos en formas trigonométricas básicas y calculamos el área de cada

Comparación de clasificadores

$$Area\ Cuadrado = Base * altura \qquad Area\ Triangulo = \frac{1}{2} * Base * altura$$

Se evalúan las instancias por parejas:

- **A 11** → Las instancias pertenecen al mismo clúster y al mismo grupo
- **D 00** → Las instancias No pertenecen al mismo clúster ni al mismo grupo
- **B 01** → Las instancias pertenecen al mismo clúster pero no al mismo grupo
- **B 10** → Las instancias pertenecen al mismo grupo pero no al mismo clúster

$$Jaccard = \frac{A}{A+B+C} \qquad Fowlkes-Mallows = \sqrt{\frac{A}{A+B} * \frac{A}{A+C}} \qquad RandIndex = \frac{A+D}{A+B+C+D}$$

- **Coeficiente de corelacion** permite medir lo similares que son dos clasificadores en función del número de aciertos y fallos que cometen.
  - Los clasificadores que más correlan son aquellos que predijeran siempre lo mismo.
    En este caso B y C serán siempre cero por lo que la ecuación quedaría como $1/A * D$ quesería siempre positivo
  - Los clasificadores que menos correlan son aquellos que predijeran siempre lo opuesto.
    En este caso A y D serian cero por lo que la ecuación quedaría como $-1/B * C$ quesería siempre negativo

		Clasificadas en el mismo Clúster	
		Si	No
Pertenecen al mismo grupo experto	Si	11 **A**	01 **B**
	No	10 **C**	00 **D**

$$coeficiente\ decorrelacion = \frac{A*D - B*C}{(A+B)*(D+C)*(A+C)*(B+D)}$$

$$Estadistica\ Q = \frac{A*D - B*C}{A*D + B*C} \qquad Desacuerdo = B + C \qquad Doble\ Falta = D$$

Orden de complejidad

$$S(N,K) = \frac{1}{K!} * \sum_{i=1}^{K} \left[(-1^K) * \binom{K}{i} * i^N \right] \quad siendo \quad \binom{K}{i} = \frac{K!}{i! * (K-i)!}$$

- N Numero de instancias - K Numero de subconjuntos
- Casos básicos
  - 1 Clúster con N instancias solo admite una agrupación. Todas las instancias en dicho clúster **S(N,1) = 1**
  - N Clusters con N instancias solo admite una agrupación. Una instancia por cada clúster **S(N,N) = 1**
- Caso general $S(N,K) \longrightarrow K * S(N-1) \quad S(N-1, K-1)$

Orden Complejidad RedNeuronal

d = inputs H = Neuronas Ocultas K = Salidas

n = Instancias de entrenamiento

[[W = (d + 1) * H] + [b = (H + 1) * K]] * n

Parametros W = Capa 0 * Capa 1 * ... * Capa i

Bias b = Capa 1 + Capa 2 + ... Capa i

Clustering Jerárquico

Asume una topología en árbol que define dependencias entre instancias

Descubre agrupamientos con un número de grupos que va desde 1 hasta N

Ofrece relación entre particiones con distinto número de clusters según una jerarquía

Cálculo de las distancias

Single Link distancia entre las instancias más cercanas

Complete Link distancia entre las instancias más lejanas

Abaraje Link distancia entre los centroides de cada clúster

- En el nivel inferior hay tantos clusters como instancias cada uno con una sola muestra
- En cada iteración se agrupan los dos clusters más cercanos anotando la distancia en la que se produjo dicha unión
- El algoritmo termina cuando solo queda un clúster

Este algoritmo ofrece una estructura en árbol que permite distintas agrupaciones en función de por donde se realice el corte

Clustering Particional Algoritmo K-Medias

Consiste en agrupar los datos en un número de conjuntos pre-determinado y posteriormente se van moviendo las instancias de un clúster a otro en base a las medidas de similitud/disimilitud. Necesita como parámetro el número de clusters a predecir

Se trata de un algoritmo de cuantificación vectorial uniforme que pretende obtener el mínimo o máximo sub optimo local de un conjunto de datos:

- Inicialmente elegimos al azar los valores que formaran parte del CodeBook seleccionando tantas instancias como clusters a predecir.
- Asignamos cada instancia a un único clúster comparando las distancias entre dicho clúster y los centroides del CodeBook
- Recorremos los elementos que han sido asignados a cada clúster con la finalidad de recalcular el centroide y actualizar el CodeBook
- Se repite el proceso hasta que la solución converja. Para ello se calcula el error cometido entre esta iteración y la anterior

Autor: Cuesta Alario David

Técnicas de Evaluación

Exploran la estructura intrínseca de los datos como compromiso de dos métricas:

$$\sum_{\forall \text{ Instancias } \epsilon \text{ Cluster}} \mathbf{d^2(\text{ Instancia , Centroide Cluster })} \qquad \sum_{\forall \textit{ Clusters}} \textit{SSE}_{\text{Clúster}}$$

Cohesión: Mide la cercanía de las instancias que pertenecen a cada clúster mediante indicadores estadísticos como:

- **SSE Clúster**: Suma de los errores cuadráticos de cada instancia respecto al centroide del clúster al que pertenecen
- **SSE Partición:** Suma de la cohesión interna de todos los clusters

Separabilidad: Mide la distancia inter-grupal que existe entre los distintos clusters

- **SSE Externo**: Suma de los errores cuadráticos de cada instancia respecto a cada uno de los centroides de los clusters a los que No pertenece dicha instancia
- **BSS:** Suma de los errores cuadráticos del centroide de cada clúster con respecto al centroide global de la partición (centroide de los centroides)

$$\text{SSE Externo: } \sum_{\forall \text{ Instancias}} \left[\sum_{\forall \text{ Clusters} \neq \text{ Cluster de la instancia}} \mathbf{d^2(\text{ Instancia , Centroide Cluster })} \right] \qquad \text{BSS: } \sum_{\forall \text{ Cluster}} \mathbf{d^2(\text{ Centroide Cluster , Centroide Particion})}$$

Silhouette

Es un indicador que combina la cohesión y la separabilidad en una única variable

$$\textbf{Silhouette} = \frac{b(Instancia) - a(Instancia)}{MAX[\,b(Instancia), a(instancia)]}$$

- a Es la distancia media entre una instancia y el resto de instancias de su mismo clúster. Un valor pequeño de a indica que la instancia analizada está cerca de las demás instancias de su clúster

$$a(Instancia) = \frac{\sum_{\forall \text{ Instancias } \epsilon \text{ Cluster}} \mathbf{d^2(}\ Instancia,\ \textbf{Resto de instancias del mismo cluster }\mathbf{)}}{Numero\ Instancias\ del\ cluster - 1}$$

- b Es distancia media entre más cercana entre la instancia y cada uno de los demás clusters. Un valor pequeño de b indica que la instancia analizada está lejos de los demás clusters (Tanto a como b deben ser siempre positivos dado que son cálculos de distancias)

$$b(Instancia) = \underset{\forall \text{ Clusters} \neq \text{ Cluster de la instancia}}{\text{MIN}} \left[\frac{\sum_{\forall \text{ Instancias } \epsilon \text{ Cluster}} \mathbf{d^2(}\ Instancia,\ \textbf{Instancia }\mathbf{)}}{Numero\ Instancias\ del\ cluste} \right]$$

$$\textbf{Silhouette}(Instancia) \longrightarrow \begin{cases} 1 - \frac{a}{b} & Si \quad a < b \\ 0 & Si \quad a = b \\ \frac{b}{a} - 1 & Si \quad a > b \end{cases}$$

Silhouette Se puede utilizar como indicador para encontrar el número óptimo de clusters. Se calcula la Silhouette para cada posible agrupación y se escoge la agrupación que la maximice.

- Esta comprendido entre 1 y -1 Si es negativo indica que la agrupación es mala (poca cohesión y/o separabilidad)

Regresión Lineal

Es bastante habitual comparar los errores con respecto de la media para para comprobar la desviación del modelo. El **coeficiente de determinación** denominado **R^2** es un estadístico que permite determinar la calidad de un modelo lineal para replicar los resultados.

$$R^2 = \frac{SSR}{SST} \qquad \textbf{SSE} = \sum_{\forall \text{ Instancias}} \textbf{(Salida Real(Instancia) – Salida Estimada(Instancia))}^2$$

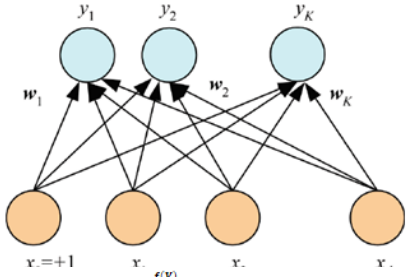
$$\textbf{SSR} = \sum_{\forall \text{ Instancias}} \textbf{(Salida Estimada(Instancia) – Media detodas las instancais)}^2$$

$$\textbf{SST} = \sum_{\forall \text{ Instancias}} \textbf{(Salida Real(Instancia) – Media detodas las instancais)}^2$$

- Su principal propósito es predecir futuros resultados o probar una hipótesis.
- Adquiere valores entre 0 y 1.

Entrenamiento del perceptrón

Consiste en buscar los parámetros internos que mejor se ajusten a la muestra utilizando como entradas los atributos de todas las instancias de entrenamiento:



- Inicializamos los pesos de cada perceptron a valores cercanos a cero
- Se analizan las instancias de una en una en orden aleatorio
  - Se calcula la salida de la red para la instancia actual y se le aplica la función de activación

$$Y = W * X \xrightarrow{f(Y)} Clase$$

  - Se calcula el error y si no es cero Se actualizan los pesos en base al error del modelo actual con respecto a la salida esperada Con la finalidad de contrarrestar el peso cada instancia para aproximar los resultados futuros a la salida deseada

$$W \longleftarrow W + \delta * (\ Salida\ Real - Salida\ Estimada\ ) * X$$

- Este proceso se repite hasta que la variación del error cruce un umbral admisible

$$Y = W * X + b \xrightarrow{f(Y)} Salida\ discreta$$

$$\begin{pmatrix} W_{11} & W_{21} & W_{31} & W_{41} \\ W_{12} & W_{22} & W_{32} & W_{42} \\ W_{13} & W_{23} & W_{33} & W_{43} \end{pmatrix} * \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} \xrightarrow{f(Y)} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$Y_i = 1 * b_i + X_1 * W_{1-i} + X_2 * W_{2-i} + \ \dots\ + X_n * W_{n-i}$$

To \ From	X_1	X_2	X_n	b
Y_1	W_{1-1}	W_{2-1}	W_{n-1}	b_1
Y_2	W_{1-2}	W_{2-2}	W_{n-2}	b_2
Y_k	W_{1-k}	W_{2-k}	W_{n-k}	b_n

Sigmoide $\mathbf{h(Y)} = \frac{1}{1+e^{-Y}}$ $\quad Y = \begin{cases} 0 & Si \quad h(Y) < 0.5 \\ 1 & Si \quad h(Y) \geq 0.5 \end{cases}$ $\quad \mathbf{h(Yi)} = \frac{e^{Yi}}{e^{Y1} + e^{Y2} + \ \dots + e^{Yi} + \dots\ + e^{Yn}}$

Tangente hiperbólica $\mathbf{h(Y)} = 1 - tanh^2(X)$

Función signo $Y = \begin{cases} 1 & Si \quad Y \geq 0 \\ -1 & Si \quad Y < 0 \end{cases}$

Asociación

Support ⟶ Probabilidad conjunta. Numero de sucesos en los que se dan X e Y al mismo tiempo

Confidence ⟶ Probabilidad condicionada. Probabilidad de que suceda Y sabiendo que ha sucedido X

Support Normalizado en función del número de apariciones de la frecuencia del antecedente

Lift ⟶ Mide la dependencia entre dos sucesos (sucesos independientes si Lift = 1)

$$\text{Support } (X \Rightarrow Y) = P(X,Y)$$

$$\text{Confidence } (X \Rightarrow Y) = P(Y|X) = \frac{P(X,Y)}{P(X)}$$

$$\text{Lift } (X \Rightarrow Y) = \frac{P(X,Y)}{P(X)P(Y)}$$

Desglosamos los ítem-sets en dos sub-conjuntos donde un sub-conjunto sea el antecedente y el otro el consecuente

Realizaremos una poda delos ítems en función del Support: Trataremos de analizar aquellos ítems que tengan suficiente Support porque si el ítem aparece pocas veces cualquier regla de asociación resultante tiene una alta probabilidad de ser fruto del azar.

- Para reducir el coste computacional se determina un umbral de modo que si no se supera se corta el proceso de análisis:
  - Si una asociación no cumple al asociar otro elemento cumplirá menos o igual
  - Si dos atributos independientes no están asociados al combinarlos lo estarán menos o igual

Después de la poda analizaremos el Confidence para seleccionar los ítems que más apariciones tengan teniendo en cuenta sus apariciones globales. Con esto tenemos el número de veces que ha aparecido el ítem XY en relación al número de veces que ha aparecido el elemento X.

Combinación de clasificadores

Se basan en los resultados de varios clasificadores débiles para obtener un clasificador más robusto.

- Da lugar a distintos modelos-base haciendo inferencia de distintos conjuntos de entrenamiento
- Dada una instancia se le asigna la clase más votada por los distintos modelos-base

La diferencia entre ambas es que Boosting asigna un peso al resultado de cada clasificador en función de su confianza.

$$\varepsilon_j \longrightarrow B_j = \frac{\varepsilon_j}{1 - \varepsilon_j} \xrightarrow{Weighted\ error} \mathbf{w}_j = Log_2\left(\frac{1}{B_j}\right) \qquad \textbf{(1 - Accuracy)}$$

Se dispone de un conjunto de L modelos-base:

- Media aritmética: $y_i(\mathbf{x}) = \frac{1}{L}\sum_{j=1}^{L} d_{ji}(\mathbf{x})$
- Máximo: $y_i(\mathbf{x}) = \max_{j=1}^{L} d_{ji}(\mathbf{x})$
- Producto: $y_i(\mathbf{x}) = \prod_{j=1}^{L} d_{ji}(\mathbf{x})$