Aprendizaje supervisado

Minería de Datos - Grado en Matemáticas

DSLAB

2025-10-26

Introducción

- Se conoce la variable objetivo que se desea predecir o clasificar
- **Variable objetivo**: contiene la información que se quiere explicar o entender en base al resto de las variables del conjunto de datos
- **Objetivo**: desarrollar modelos que capturen patrones y relaciones entre las variables con el fin de realizar predicciones precisas
  - **Clasificación**. Variable objetivo es categórica (problemas binarios o multiclase). Asignar observaciones a las diferentes categorías o clases
    - Ej: Predecir si un mail es span o no, si un paciente tiene una enfermedad
  - **Regresión**. Variable objetivo es continua. Predicciones numéricas.
    - Ej: Precio de una casa, demanda de productos

Algunos modelos

- Modelo lineal: Análisis Discriminante Lineal
- k-vecinos más próximos
- Árboles de decisión
- Métodos ensamblados (Random Forest, Bagging, Boosting)
- Naïve Bayes
- Modelos de mezcla de Gaussianas

Análisis Discriminante Lineal (ADL)

- Algoritmo de **clasificación**
  - Target: variable categórica
  - Variables explicativas continuas: distribución Normal
- **Objetivo**: Construir la combinación lineal de las variables explicativas que mejor discrimina las clases. En base a dicho hiperplano separador se clasificarán las nuevas observaciones

- Teorema de Bayes para clasificación:

$$P(C = 1|\mathbf{X} = \mathbf{x}) = \frac{f_1(\mathbf{x})P(1)}{\sum_{c=1}^{C_n} f_c(\mathbf{x})P(c)}$$

siendo $P(C = 1|\mathbf{X} = \mathbf{x})$ la probabilidad a posteriori, es decir, la probabilidad de la clase 1 dada la observación $\mathbf{x}$. $P(c)$ es la probabilidad a priori de la clase c. Nótese que $\sum_{c=1}^{C_n} P(c) = 1$. Finalmente $f_c(\mathbf{x})$ es la función de densidad condicional de las $\mathbf{X}$ en la clase c

Análisis Discriminante Lineal (ADL)

- En el caso del ADL, se asume que la densidad de cada clase es una Normal multivariante

$$f_c(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^t \Sigma_k^{-1}(\mathbf{x}-\mu_k)}$$

- También se asume que las clases comparten la misma matriz de varianzas covarianzas $\Sigma_c = \Sigma, \ \forall c$

En base a estas asunciones, comparemos las probabilidades de ambas clases:

$$log\left(\frac{P(C=1|\mathbf{X}=\mathbf{x})}{P(C=0|\mathbf{X}=\mathbf{x})}\right) = log\left(\frac{f_1(\mathbf{x})}{f_0(\mathbf{x})}\right) + log\left(\frac{P(1)}{P(0)}\right) =$$

$$log\left(\frac{P(1)}{P(0)}\right) - \frac{1}{2}(\mu_1+\mu_0)^t\Sigma^{-1}(\mu_1-\mu_0) + \mathbf{x}^t\Sigma^{-1}(\mu_1-\mu_0)$$

Obtenemos una ecuación que es lineal en $\mathbf{x}$. Es decir, la frontera de decisión entre las clases 0 y 1 es una ecuación lineal, un hiperplano en dimensión p

En la práctica, se desconocen los parámetros de la distribución Normal. Se estiman con los datos de entrenamiento:

- $\widehat{P(c)} = n_c/n$, siendo n_c el tamaño de la clase c y n el total
- $\hat{\mu}_c$ media muestral de los elementos de la clase c
- $\widehat{\Sigma}$ varianza muestral de los elementos de la clase c

Clasificación: ADL clasificará una observación como de la clase 1 si

$$\mathbf{x}^t \widehat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0) > \frac{1}{2}(\hat{\mu}_1 + \hat{\mu}_0)^t \widehat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0) - log(n_1/n_0)$$

Si llamamos $\mathbf{w} = \widehat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0)$, podemos reescribir lo anterior como

$$\mathbf{x}^t \mathbf{w} > \frac{1}{2}(\hat{\mu}_1 + \hat{\mu}_0)^t \mathbf{w} - log(n_1/n_0)$$

La frontera de decisión entre ambas clases es

$$\mathbf{x}^t \mathbf{w} = \frac{1}{2}(\hat{\mu}_1 + \hat{\mu}_0)^t \mathbf{w} - log(n_1/n_0)$$

que es una combinación lineal de las variables explicativas

Interpretación: Proyectar el punto a clasificar y las medias de las clases en una recta y asignar la observación a la clase con media más cercana

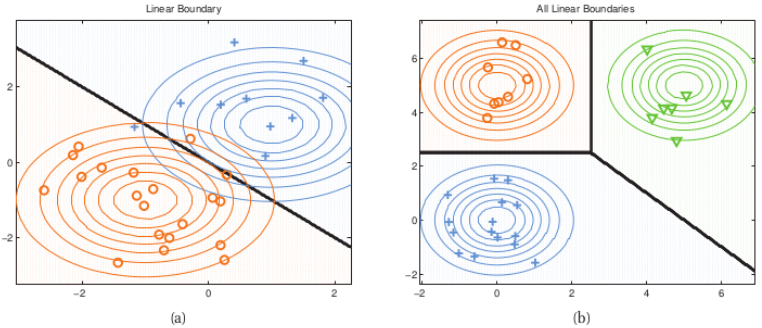


Figure 4.5 Linear decision boundaries in 2D for the 2 and 3 class case. Figure generated by `discrimAnalysisDboundariesDemo`.

Figure 1: Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

Análisis Discriminante Lineal: ventajas y desventajas

- **Ventajas**
  - Simple y rápido
  - Eficiente cuando se cumple las hipótesis
  - Clasificación de observaciones en grupos determinados –> facilita la interpretación
  - Combinación de información para la frontera de decisión
- **Desventajas**
  - Asume normalidad e igualdad de varianzas
  - Sensible a outliers
  - Es un clasificador lineal
  - Requiere cierto tamaño de muestra

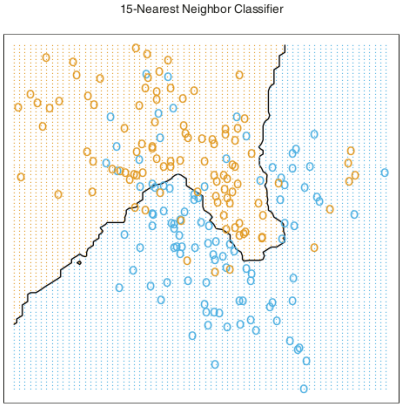
k-vecinos más próximos (k-nn)

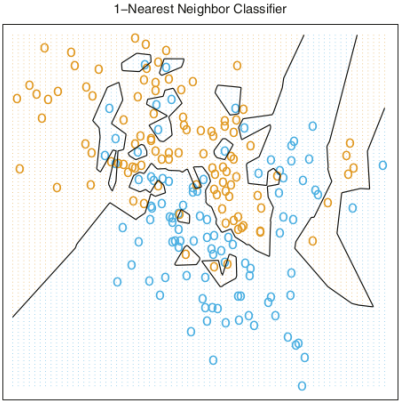
- k-nn: k nearest neighbors
- Se basa en la noción de similitud o distancia entre individuos, en la idea de que observaciones similares se encuentran próximas
- Desafíos:
  - Métrica de similaridad utilizada para evaluar el parecido entre observaciones
  - Noción de cercanía: ¿Qué k elegir? –> Podemos usar técnicas como validación cruzada y grid search
- Funcionamiento:
  - Clasificación: devuelve la clase predominante entre los k vecinos (la moda)
  - Regresión: devuelve la media de la variable respuesta en los k vecinos

k-vecinos más próximos (k-nn)

- **Ventajas**
  - Sencillo y de fácil implementación
  - No asume una distribución específica sobre los datos
  - Adaptabilidad a datos cambiantes
  - Interpretabilidad
  - Robusto frente al ruido
  - Sirve para regresión y clasificación (tanto binaria como multiclase)
- **Desventajas**
  - Sensible a la elección de k y de la métrica
  - Coste computacional en alta dimensionalidad
  - Datos desbalanceados: sesgo hacia la clase mayoritaria

k-vecinos más próximos (k-nn)





Origen de las imágenes: Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.

Árbol de decisión

- Algoritmos de ML basados en la información
- Determinan qué variables explicativas proporcionan la mayor **ganancia de información** para medir la variable objetivo
- Proceso de particionamiento en el conjunto de variables explicativas
  - **Nodo raíz**: Nodo origen, todas las observaciones forman parte
  - **Nodos internos**: Nodos que se crean al definir reglas sobre una variable explicativa
  - **Nodos hojas**: Nodos terminales del árbol (donde se realiza la clasificación)

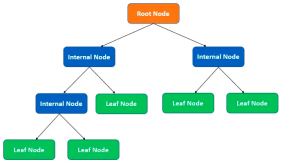


Figure 2: https://iprathore71.medium.com/complete-guide-to-decision-tree-cee0238128d

Árbol de decisión

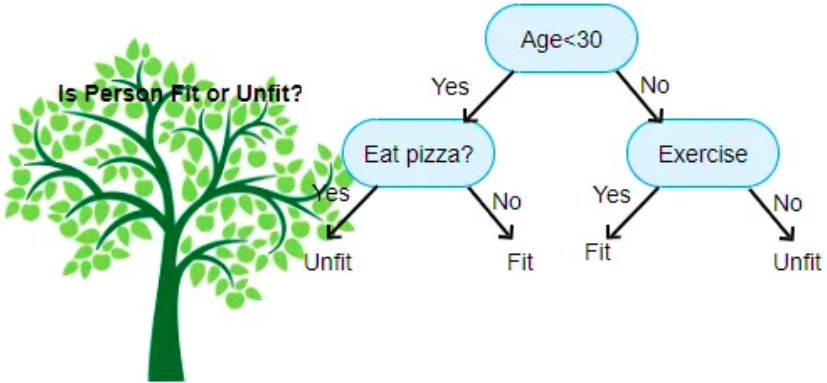


Figure 3: https://iprathore71.medium.com/complete-guide-to-decision-tree-cee0238128d

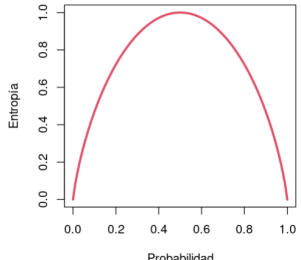
Árbol de decisión

- Las variables que más discriminen las clases están en la parte superior del árbol
- ¿Cómo evaluar esta discriminación? –> Métricas
  - Entropía
  - Índice de Gini

Árbol de decisión: Entropía

- La **entropía** es una medida teórica de la "*incertidumbre*" contenida en un conjunto de datos
- La **entropía** de un conjunto de n valores distintos, igualmente probables, es el menor número de preguntas de *sí/no* necesarias para determinar un valor desconocido extraído de las posibilidades:

$$Entropa = -\sum_{c=1}^{C} p_c \log_2(p_c)$$



Procedimiento del árbol usando la entropía

1. Calcular la **entropía** del conjunto original de datos
2. Para cada variable explicativa, se crean los conjuntos resultantes **dividiendo las observaciones** en el conjunto de datos utilizando un umbral para dicha variable. Se calcula la entropía de cada nodo individual de división y la media ponderada de todos los nodos hijos disponibles en una división
3. Calcular la **Ganancia de Información** restando la entropía restante (paso 2) del valor de entropía original (paso 1)

Para cada nodo en el árbol, se elige para la división aquella variable que maximiza la ganancia de información

Árbol de decisión: Índice de Gini

- Mide la pureza de los nodos
- El índice de Gini es el error esperado. p_c es la probabilidad de que una entrada cualquiera de la hoja pertenezca a la clase c y $(1 - p_c)$ es la probabilidad de que sea erróneamente clasificada:

$$Gini = \sum_{c=1}^{C} p_c(1 - p_c) = 1 - \sum_{c=1}^{C} p_c^2$$

  siendo C es el total de clases
- Comportamiento similar a la entropía
- Buscamos que la impureza de los nodos vaya disminuyendo –> índice de Gini bajo

Árbol de regresión: reducción de la varianza

- Reducción de la varianza: equivalente de la entropía o el índice de Gini para árboles de regresión
- Se utiliza la varianza para encontrar la mejor división
- Se calcula la varianza de cada división como la media ponderada de la varianza de cada nodo resultado de dicha división

Árbol de decisión: criterio de parada

- Error 0 de clasificación –> ¿Qué implicaría esto?
- Profundidad específica
- Número mínimo de observaciones en el nodo
- Cantidad de mejora

Árbol de decisión: ventajas

- Explicabilidad
- Compatibilidad con datos mixtos
- No requiere escalado de variables
- Manejo de datos faltantes
- Robustez ante valores atípicos
- Captura de no linealidades
- Eficiencia computacional

Árbol de decisión: desventajas

- Sensibilidad a cambios en los datos. Pequeños cambios en los datos pueden implicar un árbol diferente
- Naturaleza jerárquica –> propagación del error
- Propensión al sobreajuste
- Dificultad para modelar relacionales lineales
- No son óptimos para datos de alta dimensión
- Difícil interpretación cuando los árboles son muy profundos

Métodos de ensamblado

- **La unión hace la fuerza**: se basa en la idea de que la unión de múltiples modelos puede mejorar significativamente el rendimiento de predicción en comparación con un solo modelo
- Interpetación del Machine Learning de la **sabiduría colectiva**
- Clave: **diversidad** entre los distintos modelos (entrenando con distintas muestras del conjunto de datos o con distintos conjuntos de variables)
- Los más famosos:
  - Bagging
  - Boosting
  - Random Forest

Bagging

- **Bagging (Bootstrap Aggregating)**: técnica diseñada para mejorar la precisión y estabilidad de los modelos predictivos. Combina una serie de modelos "débiles"
- **Algoritmo**:
  1. **Bootstrap**: Se obtienen m muestras con reemplazamiento de tamaño n (tamaño real del conjunto de datos). Estas serán las muestras de entrenamiento
  2. **Modelo base**: Se entrenan m modelos (partiendo de un modelo base como un DT) usando las m muestras de entrenamiento
  3. **Predicciones**: Se obtienen un total de m predicciones del conjunto de test, 1 por cada modelo
  4. **Combinación**: Se combinan las predicciones de los m modelos para dar una predicción final
     - Clasificación: regla del voto mayoritario
     - Regresión: media

Bagging

- **Ventajas**
  - Reducción de la varianza. Más robusto y menos propenso al sobreajuste
  - Mayor precisión
  - Estabilidad
  - Mayor capacidad de generalización
- **Desventajas**
  - Mayor complejidad computacional
  - Menor interpretabilidad
  - No garantiza la mejora
  - Menos efectivo con modelos base inestables

Random Forest

- Bosque formado por múltiples Árboles de Decisión
- Los distintos árboles se entrenan con un subconjunto aleatorio de observaciones (Bagging) y también un subconjunto aleatorio de variables –> Diversidad
- Cada árbol individual no es potente, pero la combinación de todos, que han aprendido cosas distintas, sí resulta potente
- Ofrece información sobre la importancia de las variables
- ¿Ventajas y desventajas?

Boosting

- Se centra en mejorar iterativamente un modelo "débil"
- **Algoritmo**
  1. Se entrena un modelo base en el conjunto de datos original
  2. Se evalúa su rendimiento. Se da más peso a las observaciones clasificadas erróneamente
  3. Se entrena otro modelo débil usando los pesos del paso previo
  4. Se combina la predicción de los modelos base ponderando sus predicciones en función de su rendimiento. Los mejores modelos tendrán más peso en la predicción final.
- Cada modelo nuevo se centra en corregir las deficiencias del previo
- La combinación final es un ensamblado fuerte
- Algoritmos Boosting populares: AdaBoost, Gradient Boosting, XGBoost

Boosting

- **Ventajas**
  - Mejora del rendimiento
  - Reducción del sesgo
  - Gestión de datos desequilibrados
  - Capacidad para detectar patrones complejos
- **Desventajas**
  - Coste computacional
  - Tiempo de entrenamiento (secuencial)
  - Menor interpretabilidad
  - Ajuste de hiperparámetros

Bagging vs Boosting

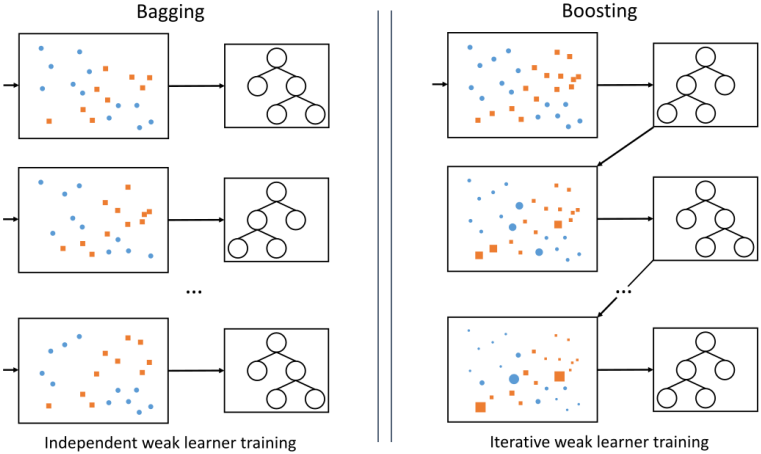


Figure 4: Imagen de González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. Information Fusion, 64, 205-237.

Naive Bayes

- **Clasificador ingenuo de Bayes** (en inglés "*Naïve Bayes*") es un clasificador sencillo que se basa en el conocido **teorema de Bayes**
- Teorema de Bayes: relaciona la probabilidad condicional de dos eventos A y B

$$P(A \cap B) = P(A, B) = P(A)P(B|A) = P(B)P(A|B) \Rightarrow$$
$$\Rightarrow P(B|A) = \frac{P(B)P(A|B)}{P(A)}$$

Aplicación en un problema de clasificación:

Supongamos $X_1, X_2, \dots, X_n$ variables explicativas independientes dado el valor de la variable objetivo Y_k. Es decir,

$$P(X_1|X_2, \dots, X_n, Y_k) = P(X_1|Y_k); \quad P(X_2|X_3, \dots, X_n, Y_k) = P(X_2|Y_k); \quad \dots$$

Aplicando el teorema de Bayes recursivamente

$$P(X_1, X_2, \dots, X_n, Y_k) = P(Y_k)P(X_n|Y_k)P(X_{n-1}|Y_k) \cdots P(X_1|Y_k)$$

Por tanto

$$P(Y_k|X_1, \dots, X_n) = \frac{P(Y_k)\prod_{j=1}^{n} P(X_j|Y_k)}{P(X_1, X_2, \dots, X_n)}$$

Naive Bayes

Fórmula:

$$P(Y_k|X_1, \dots, X_n) = \frac{P(Y_k) \prod_{j=1}^{n} P(X_j|Y_k)}{P(X_1, X_2, \dots, X_n)}$$

Componentes:

- **Denominador**: constante
- $P(Y_k)$: probabilidad a priori
- **Verosimilitud** $\prod_{j=1}^{n} P(X_j|Y_k)$: mide cómo de verosímiles son las variables explicativas dado Y_k
  - Variable cuantitativa: Normal
  - Variable cualitativa: proporción
- **Predicción**: clase con mayor $P(Y_k|X_1, \dots, X_n)$

Mezcla de Gaussianas

- **GMM: Gaussian Mixture Models**
- Se basan en la idea de que un conjunto de datos está compuesto por varias distribuciones gaussianas (normales)
- GMM permiten descomponer un conjunto de datos en múltiples componentes gaussianas, cada una de las cuales describe una parte de la distribución de los datos
- Cada componente gaussiana representa una subdistribución de datos y se caracteriza por su media (promedio) y desviación estándar (dispersión)
- Estimación de los parámetros: algoritmos de optimización para maximizar la probabilidad conjunta de los datos observados
- Cada observación se asigna a la gaussiana a la que pertenece con mayor probabilidad
- Sensibles al número de componentes y la inicialización

Referencias

González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion*, *64*, 205-237.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Kelleher, John D, y Brendan Tierney. 2018. *Data science*. MIT Press.

Zhou, Z. H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.