TRABAJO PRÁCTICO N°5 – Sistema de Soporte para la Toma de Decisiones

Acosta, Luis Matías.
Mazzaglia, Ian.
Murua, Federico.

1.1. Preprocesamiento de datos

1.Acciones básicas

- Mean subtraction
  - Si los datos siguen la misma distribución, entonces se puede considerar restar el valor promedio para cada caso.
  - Por ejemplo, en Una imagen, ésta normalización tiene la propiedad de eliminar el brillo promedio de un punto específico (eliminación del valor píxel promedio). se aplica tanto imágenes en color o monocromática
- Normalización:
  - Significa aplicar una transformación para que los datos transformados se distribuyan normalmente.
  - Es un “rescaling” del rango original de los datos dando como resultado todos los valores dentro del rango 0 y 1.
  - Puede ser útil e incluso necesaria en algunos algoritmos de aprendizaje automático cuando los datos de series de tiempo tienen valores de entrada con escalas diferentes y otros algoritmos como el de k-Vecinos más cercanos que usa cálculos de distancia, regresión lineal y Redes neuronales artificiales.
  - Requiere que se sepa o pueda estimar con precisión los valores observables mínimos y máximos a partir de los datos disponibles.
- Scaling:
  - Es una serie de elementos de la misma especie, ordenados gradualmente en función de alguna de sus características o cualidades.
  - Por ejemplo, convertir datos dados en milímetros a metros porque es más conveniente.

2. Análisis de Componentes Principales ACP/PCA

En estadística, el análisis de componentes principales (en español ACP, en inglés, PCA) es una técnica utilizada para describir un conjunto de datos en términos de nuevas variables ("componentes") no correlacionadas. Los componentes se ordenan por la cantidad de varianza original que describen, por lo que la técnica es útil para reducir la dimensionalidad de un conjunto de datos.
Técnicamente, el ACP busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. Esta convierte un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables sin correlación lineal llamadas componentes principales.

Ejemplo: Un set de datos puede describir la altura y el peso de 100 niños entre 2 y 15 años. Ambas variables están, obviamente, correlacionadas (los niños de más edad son más altos y pesan más). El análisis de componentes principales describe los datos en términos de dos nuevas variables. El primer componente se puede interpretar como "tamaño" o "edad" y recoge la mayor parte de la varianza de los datos originales. El segundo componente describe variabilidad en los datos que no está correlacionada en absoluto con el primer componente principal "tamaño", y (probablemente) sea difícil de interpretar. Si el objetivo es reducir la dimensionalidad de los datos, se puede descartar este segundo componente principal.

Incrustacion Estocastica de vecinos distribuidos en T t-SNE

t-SNE es un algoritmo de aprendizaje automático para la visualización, desarrollado por Laurens van der Maaten y Geoffrey Hinton. Es una técnica de reducción de dimensionalidad no lineal adecuada para incorporar datos de alta dimensión para visualización en un espacio de baja dimensión de dos o tres dimensiones. Específicamente, modela cada objeto de alta dimensión por un punto bidimensional o tridimensional de tal manera que objetos similares se modelan por puntos cercanos y objetos diferentes se modelan por puntos distantes con alta probabilidad.

El algoritmo t-SNE comprende dos etapas principales. Primero, t-SNE construye una distribución de probabilidad sobre pares de objetos de alta dimensión de tal manera que los objetos similares tienen una alta probabilidad de ser recogidos, mientras que los puntos diferentes tienen una probabilidad extremadamente pequeña de ser recogidos. En segundo lugar, t-SNE define una distribución de probabilidad similar sobre los puntos en el mapa de baja dimensión, y minimiza la divergencia Kullback-Leibler entre las dos distribuciones con respecto a las ubicaciones de los puntos en el mapa.

t-SNE se ha utilizado para la visualización en una amplia gama de aplicaciones, incluida la investigación de seguridad informática, análisis de música, investigación del cáncer, bioinformática, entre otras.

La Aproximación y Proyección Uniforme del Múltiple UMAP

Es una técnica de reducción de dimensión que se puede usar para la visualización de manera similar a t-SNE, pero también para la reducción general de la dimensión no lineal. El algoritmo se basa en tres supuestos sobre los datos:

- Los datos se distribuyen uniformemente en una variedad riemanniana;
- La métrica riemanniana es localmente constante (o puede ser aproximada como tal);
- El colector o tubo está conectado localmente.

A partir de estas suposiciones, es posible modelar la variedad con una estructura topológica difusa. La incrustación se encuentra al buscar una proyección dimensional baja de los datos que tiene la estructura topológica difusa equivalente más cercana posible.

ventajas:

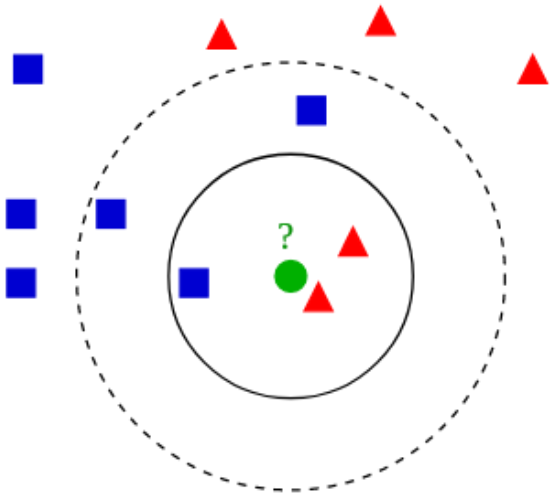
- Puede manejar grandes conjuntos de datos y datos de alta dimensión sin demasiada dificultad, escalando más allá de lo que la mayoría de los paquetes t-SNE pueden administrar.
- Puede usarse como una técnica de reducción de dimensión de propósito general, como un paso preliminar para otras tareas de aprendizaje automático.
- A menudo se desempeña mejor en la preservación de aspectos de la estructura global de los datos que t-SNE. Esto significa que generalmente puede proporcionar una mejor visión de "gran imagen" de sus datos, así como preservar las relaciones vecinas locales.
- admite una amplia variedad de funciones de distancia, incluidas las funciones de distancia no métricas, como la distancia del coseno y la distancia de correlación.

1.2 Introducción a Aprendizaje Automatico

Algoritmo de Clasificación Supervisado: K Vecinos más cercanos

El algoritmo de los k vecinos más cercanos (k-NN Nearest Neighbour) es un sistema de clasificación supervisado basado en criterios de vecindad. Los sistemas de clasificación supervisados son aquellos en los que, a partir de un conjunto de ejemplos clasificados (conjunto de entrenamiento), intentamos asignar una clasificación a un segundo conjunto de ejemplos. En particular, k-NN se basa en la idea de que los nuevos ejemplos serán clasificados a la clase a la cual pertenezca la mayor cantidad de vecinos más cercanos del conjunto de entrenamiento más cercano a él.

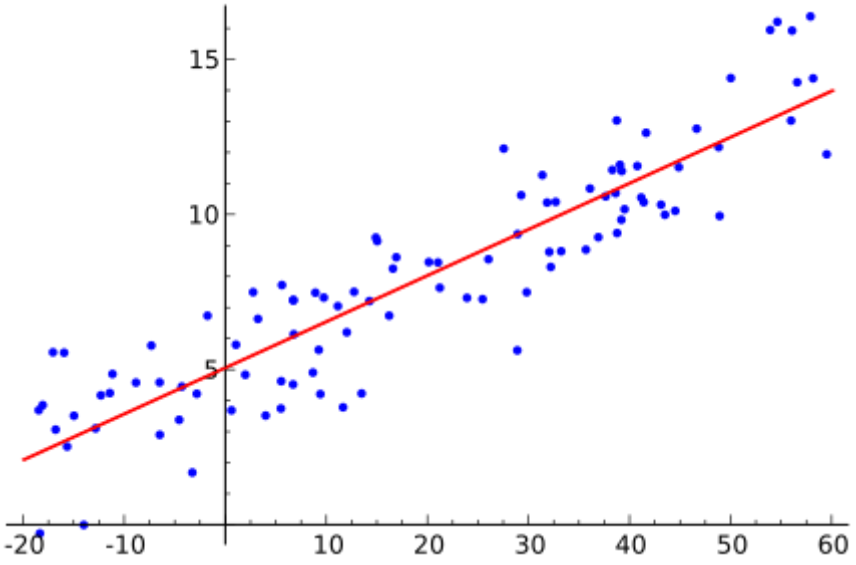
El algoritmo del vecino más cercano explora todo el conocimiento almacenado en el conjunto de entrenamiento para determinar cuál será la clase a la que pertenece una nueva muestra, pero únicamente tiene en cuenta el vecino más próximo a ella, por lo que es lógico pensar que es posible que no se esté aprovechando de forma eficiente toda la información que se podría extraer del conjunto de entrenamiento.



Con el objetivo de resolver esta posible deficiencia surge la regla de los k vecinos más cercanos (k-NN), que es una extensión en la que se utiliza la información suministrada por los k ejemplos del conjunto de entrenamiento más cercanos.
En problemas prácticos donde se aplica esta regla de clasificación se acostumbra tomar un número k de vecinos impar para evitar posibles empates (aunque esta decisión solo resuelve el problema en clasificaciones binarias). En otras ocasiones, en caso de empate, se selecciona la clase que verifique que sus representantes tengan la menor distancia media al ejemplo que se está clasificando. En última instancia, si se produce un empate, siempre se puede decidir aleatoriamente entre las clases con mayor representación.

Regresión Lineal

La regresión lineal es quizás el método más conocido para “predecir” el comportamiento de los datos o intentar hacerlo. Es considerada como una de las técnicas centrales del aprendizaje supervisado.
Siguiendo las ideas de ML siempre se tienen datos de entrada X y de salida Y, el método que se usa para analizar los datos, es la regresión lineal cuando los datos de salida son “números reales“, como 1.2, 3.1415, 2.768, etc.
Pero la técnica no solo se estudia en ML, es de los principales temas que se enseñan en estadística, econometría o en ciencias sociales y políticas.
La idea es poder ajustar los datos experimentales a una curva, no se limita a una recta, en el peor de los casos puede ser una curva más elaborada, por ejemplo polinomios. El problema de la regresión y en general de ML, es que solo se tienen las parejas de datos (x,y) y la cuestión es definir una función f(x) donde los valores sean muy cercanos a Y.
En el caso de la regresión lineal, la función f(x)=ax+b es la que se considera “mejor” para describir la relación y el siguiente problema que surge es cómo elegir “b” y “a” de modo que entre todas nuestras posibles rectas sea la “mejor” para el modelo.
Entonces el problema se reduce a elegir un modelo y estimar sus parámetros, pero teniendo una estrategia para discernir entre los que serán los mejores parámetros.



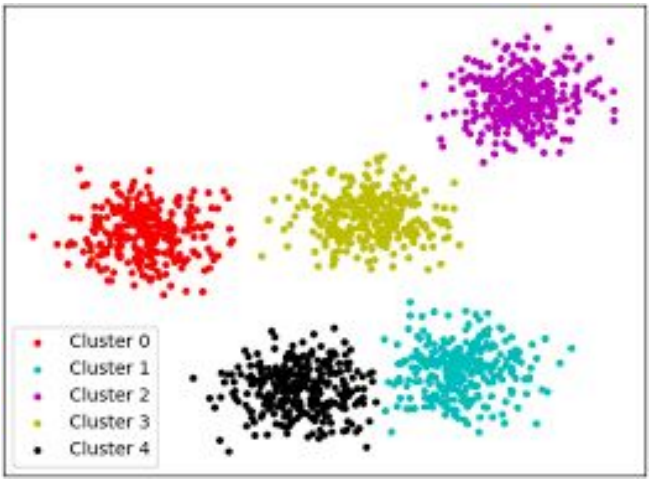
En medicina, las primeras evidencias relacionando la mortalidad con el fumar tabaco vinieron de estudios que utilizaban la regresión lineal. Los investigadores incluyen una gran cantidad de variables en su análisis de regresión en un esfuerzo por eliminar factores que pudieran producir correlaciones espurias.
En el caso del tabaquismo, los investigadores incluyeron el estado socio-económico para asegurarse que los efectos de mortalidad por tabaquismo no sean un efecto de su educación o posición económica. No obstante, es imposible incluir todas las variables posibles en un estudio de regresión

Clustering

Un algoritmo de agrupamiento (en inglés, clustering) es un procedimiento de agrupación de una serie de vectores de acuerdo con un criterio. Esos criterios son por lo general distancia o similitud. La cercanía se define en términos de una determinada función de distancia, como la euclídea, aunque existen otras más robustas o que permiten extenderla a variables discretas. La medida más utilizada para medir la similitud entre los casos es la matriz de correlación entre los nxn casos. Sin embargo, también existen muchos algoritmos que se basan en la maximización de una propiedad estadística llamada verosimilitud.

Generalmente, los vectores de un mismo grupo (o clústers) comparten propiedades comunes. El conocimiento de los grupos puede permitir una descripción sintética de un conjunto de datos multidimensional complejo. De ahí su uso en minería de datos. Esta descripción sintética se consigue sustituyendo la descripción de todos los elementos de un grupo por la de un representante característico del mismo.

En algunos contextos, como el de la minería de datos, se lo considera una técnica de aprendizaje no supervisado puesto que busca encontrar relaciones entre variables descriptivas pero no la que guardan con respecto a una variable objetivo.



Las técnicas de agrupamiento encuentran aplicación en diversos ámbitos.

- En biología para clasificar animales y plantas.
- En medicina para identificar enfermedades.
- En marketing para identificar personas con hábitos de compras similares.
- En teoría de la señal pueden servir para eliminar ruidos.
- En biometría para identificación del locutor o de caras.

Variabilidad en PCA

Para poder explicar la variabilidad del 70% de los datos es necesario utilizar 2 componentes, como se puede apreciar en la siguiente imagen.

```
[1]: import numpy as np

[2]: rng = np.random.RandomState(1)
     X = np.dot(rng.rand(2, 2), rng.randn(2, 200)).T

[3]: from sklearn.decomposition import PCA
     pca = PCA(n_components=2)
     pca.fit(X)

[3]: PCA(copy=True, iterated_power='auto', n_components=2, random_state=None,
       svd_solver='auto', tol=0.0, whiten=False)

[4]: pca.components_

[4]: array([[-0.94446029, -0.32862557],
            [-0.32862557,  0.94446029]])

[5]: pca.explained_variance_

[5]: array([0.7625315, 0.0184779])
```

Pipeline en SVM

Pipeline se puede utilizar para encadenar estimadores múltiples en uno. Esto es útil ya que a menudo hay una secuencia fija de pasos en el procesamiento de los datos, por ejemplo, selección de funciones, normalización y clasificación. Pipeline tiene dos propósitos aquí:

- Comodidad y encapsulación: Sólo debe establecer el fit y predict una vez en sus datos para que se ajuste a toda una secuencia de estimadores.
- Selección conjunta de parámetros: Puede buscar en la grilla los parámetros de todos los estimadores en la tubería a la vez.
- La seguridad: Las tuberías ayudan a evitar la filtración de estadísticas de los datos de prueba al modelo entrenado en la validación cruzada, asegurando que las mismas muestras se utilizan para entrenar a los transformadores y predictores.

Todos los estimadores en una tubería, excepto el último, deben ser transformadores (es decir, deben tener un método de transform). El último estimador puede ser de cualquier tipo (transformador, clasificador, etc.).

El Pipeline se construye utilizando una lista de pares (key, value) , donde la key es una cadena que contiene el nombre que desea dar a este paso y el value es un objeto estimador:

```
>>> from sklearn.pipeline import Pipeline
>>> from sklearn.svm import SVC
>>> from sklearn.decomposition import PCA
>>> estimators = [('reduce_dim', PCA()), ('clf', SVC())]
>>> pipe = Pipeline(estimators)
>>> pipe
Pipeline(memory=None,
         steps=[('reduce_dim', PCA(copy=True,...)),
                ('clf', SVC(C=1.0,...))])
```

1.3 Regresión

1 Describa dos métodos de regresión a elección

Regresión Lineal: Se utiliza para estimar los valores reales (costos de viviendas, número de llamadas, ventas totales, etc.) basados en variables continuas. La idea es tratar de establecer la relación entre las variables independientes y dependientes por medio de ajustar una mejor línea recta con respecto a los puntos. Esta línea de mejor ajuste se conoce como línea de regresión y está representada por la siguiente ecuación:

$$Y=\beta 0+\beta 1X1+\beta 2X2+...+\beta nXn$$

Regresión Logística: Los modelos lineales, también pueden ser utilizados para clasificaciones; es decir, que primero se ajusta el modelo lineal a la probabilidad de que una cierta clase o categoría ocurra, y a luego, utilizar una función para crear un umbral en el cual se especifica el resultado de una de estas clases o categorías. la función que se utiliza es:

$$f(x)=11+e-1$$

1.4 Clustering

1, Describa de los siguientes métodos de clustering, sus casos de uso y su escalabilidad en la implementación.

A. K-Means
B. Mean-Shift
C. Gaussian Mixtures

K-means es un método de agrupamiento o clustering, que tiene como objetivo la partición de un conjunto de n observaciones o muestras en **k** grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano.

Este algoritmo se puede aplicar a un gran conjunto de datos por lo cual ha sido ampliamente usado en muchas áreas como:

- Segmentación de Mercados.
- Visión por Computadoras.
- Geoestadística.
- Astronomía
- Minería de datos en agricultura.

Mean-Shift es un método de agrupamiento o clustering iterativo no paramétrico. Encuentra las modas de unas distribuciones, pero sin necesitar saber cuántas modas tenemos. Considera que el espacio de datos es una función de densidad de probabilidad muestreada.

Para encontrar la moda más cercana de cada punto del conjunto de datos, define una región alrededor de ese punto y encuentra su media, cambiando la situación de la media actual a la nueva (shift). Repite el proceso hasta que converja.
Debido al procedimiento iterativo costoso y estimación de densidad, mean-shift es normalmente más lento que k-means.
Su mayor aplicación suele ser el procesamiento de imágenes

Gaussian Mixtures es un método de agrupamiento o clustering que, a diferencia de K-means, permite generar cluster de una forma distinta a la circular. Además, este tipo de método de agrupamiento, permite que un ejemplo del conjunto de datos pueda pertenecer a dos o más cluster's.

2. Llevar una imagen cualquiera (recibida por línea de comando) a 80 colores con el método de *clustering K-Means* e imprima los porcentajes de estos colores en la imagen resultante. Qué aplicación le daría a este algoritmo?

Resolución del punto en el Notebook "Clustering.ipynb"

3. Aplicar el método de Mean-Shift sobre el *dataset* IRIS.

Resolución del punto en el Notebook "Clustering.ipynb"

1.5 Clasificación

1. Dos kernel posibles para una SVM

Kernel polinomial: El hecho de añadir funciones polinomiales es muy sencillo de implementar. Pero un grado polinomial bajo no puede lidiar con conjuntos de datos complejos y con un grado polinomial alto se crea una cantidad de características, haciendo que el modelo sea demasiado lento. En estas situaciones se usa un kernel polinomial para evitar este problema. Tiene el siguiente formato:

$$k(x,y) = x^T y + 1)^d$$

donde d es el grado del polinomio.

Kernel RBF Gaussiano: (Función de base radial) es del siguiente formato:

$$k(x,y) = e^{-g\|x-y\|^2}, g > 0$$

1.6 Métodos de Ensambles

El objetivo de los métodos de ensambles es combinar las predicciones de varios estimadores base construidos con un algoritmo de aprendizaje dado para mejorar la generalización/robustez sobre un solo estimador.

Generalmente se distinguen dos familias de métodos de ensamblaje:

- En los **métodos de promediación**, el principio de conducción es construir varios estimadores de forma independiente y luego promediar sus predicciones. En promedio, el estimador combinado suele ser mejor que cualquiera de los estimadores de base única porque se reduce su varianza.
- Por el contrario en los **métodos de empuje**, los estimadores de base se construyen secuencialmente y se intenta reducir el sesgo del estimador combinado. La motivación es combinar varios modelos débiles para producir un conjunto poderoso.

1.7 Cuantificación de la calidad de las predicciones

Mean Squared Error: La función mean_squared_error calcula el error cuadrático medio, una métrica de riesgo que corresponde al valor esperado del error o pérdida al cuadrado (cuadrático).

R2 Score: La función r2_score calcula R^2, el coeficiente de determinación. Proporciona una medida de qué tan probable es que el modelo prediga futuras muestras. La mejor puntuación posible es 1.0 y puede ser negativa (porque el modelo puede ser arbitrariamente peor). Un modelo constante que siempre predice el valor esperado de y, sin tener en cuenta las características de entrada, obtendría una puntuación de 0.0.

F1 Score: El puntaje de F1 puede interpretarse como un promedio ponderado de la precisión y el recall donde un puntaje de F1 alcanza su mejor valor en 1 y el peor puntaje en 0. La contribución relativa de la precisión y el recuerdo al puntaje de F1 son iguales.

Log Loss: Utilizada en la regresión logística (multinomial) y sus extensiones, como las redes neuronales, definidas como la probabilidad de registro negativa de las etiquetas verdaderas dadas las predicciones de un clasificador probabilístico. La pérdida de registro solo se define para dos o más etiquetas.

Accurancy: Mide la precisión de la predicción del modelo, por ejemplo:

```
>>> import numpy as np
>>> from sklearn.metrics import accuracy_score
>>> y_pred = [0, 2, 1, 3]
>>> y_true = [0, 1, 2, 3]
>>> accuracy_score(y_true, y_pred)
0.5
```

Precisión: La curva de recuperación de precisión muestra el equilibrio entre la precisión y la recuperación para un umbral diferente. Un área alta debajo de la curva representa tanto la recuperación alta como la alta precisión, donde la alta precisión se relaciona con una tasa de falsos positivos baja y la recuperación alta se relaciona con una tasa de falsos negativos baja. Las puntuaciones altas para ambos muestran que el clasificador está devolviendo resultados precisos (alta precisión), así como una mayoría de todos los resultados positivos (recordación alta).

Compute area under the Curve: Esta es una función general, dados los puntos en una curva para calcular el área bajo la misma.

Rand index adjusted for chance: El método Rand index calcula una medida de similitud entre dos agrupamientos, considerando todos los pares de muestras y contando los pares

que se asignan en los mismos o diferentes agrupamientos en los agrupamientos predichos y verdaderos.

1.8 Bibliografía

https://relopezbriega.github.io/blog/2015/10/10/machine-learning-con-python/
http://queirozf.com/entries/feature-scaling-quick-introduction-and-examples-using-scikit-learn#simple-feature-recaling
http://ufldl.stanford.edu/wiki/index.php/Data_Preprocessing
http://datos.gob.ar/
https://umap-learn.readthedocs.io/en/latest/
http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60
https://es.wikipedia.org/wiki/K-means
https://rua.ua.es/dspace/bitstream/10045/17323/6/segmentacion.pdf
https://es.wikipedia.org/wiki/An%C3%A1lisis_de_grupos
https://en.wikipedia.org/wiki/Mean_shift
https://www.youtube.com/watch?v=qMTuMa86NzU
https://www.youtube.com/watch?v=0NMC2NfJGqo