Buenos días, soy Daniel López y voy a presentar mi trabajo "Distribución de puntos en la esfera.Competición en Kaggle." Como se puede apreciar el trabajo consta de 2 partes, la primera de ellas, la matemática, en la que estudiaremos la obtención de distribuciones de puntos sobre la esfera que nos proporcionan buenas propiedades geométricas para aproximación. La segunda parte es la parte informática, en la que describiremos el proceso seguido durante la participación en una competición de machine learning en la plataforma Kaggle.

Distribucion de puntos en la esfera.

Objetivos

En esta parte, queremos determinar conjuntos de puntos sobre la esfera, que tengan buenas propiedades para interpolación,aproximación e integración sobre la esfera. La utilidad de esto reside en,... Finalmente, realizaremos una visualización de los puntos obtenidos en el caso particular de dimensión 3.

En primer lugar, vamos a construir el espacio de polinomios armónicos esféricos. A continución obtendremos el gradiente de dichos polinomios y calcularemos sus puntos críticos. Finalmente, veremos un resultado de integración numérica del que podremos deducir que el conjunto de puntos obtenido ofrece las propiedades que buscamos.

Esféricos Armónicos.

Para construir el espacio de armónicos esféricos necesitaremos algunas definiciones.

Vamos a considerar el espacio de los polinomios homogeneos de grado n en d dimensiones. Cabe recordar, que un polinomio homogéneo es un polinomio en que cada uno de sus términos (monomios) tienen el mismo grado; o sus elementos son de la misma dimensión. Aquí se pueden ver algunos ejemplos.

Por otro lado, se dice que una función es armónica si tiene derivadas parciales continuas de primer y segundo orden y satisfacen la ecuación de Laplace. Es decir, la suma de sus parciales de segundo orden es nula.

Uniendo estos conceptos, definimos Y_n(Rd) como el espacio de los polinomios homogeneos de grado n en Rd que son armónicos.

Finalmente, definimos el espacio de armonicos como la restriccion a la esfera del espacio Y_n.

Una vez construido el espacio el siguiente teorema nos da una base ortogonal del espacio.

Calculo del gradiente en dimensión 3

Suponemos ahora el caso particular de dimensión tres. Tomando d=3 en la expresión anterior y realizando el cambio a coordenadas polares, obtenemos la siguiente base ortogonal del espacio.

Derivando respecto a x1,x2,x3 obtenemos la siguiente expresion de las parciales. Ahora queremos, obtener los puntos que anulan el gradiente.Es decir, los puntos que anulan simultaneamente estas expresiones. Para ello igualamos a 0 estas expresiones, y obtenemos que debe de verificarse alguna de estas condiciones. De la primera se obtienen el polo sur y el polo norte de la esfera. %aqui hay que meter bien las condiciones

Ahora os voy a mostrar una simulación que hemos realizado para mostrar los puntos que se obtienen. Vamos a tomar el espacio de grado 20 y vamos a ver los puntos que se obtienen para los distintos k (k=0,. . . ,19).

Integracion numerica

Finalmente, vamos a obtener un resultado sobre integración numérica. Suponiendo el caso anterior tenemos N nodos(puntos obtenidos) y una funcion f, y los valores aproximados de cada nodo. Queremos obtener la siguente intregral. Para ello haremos uso de tecnicas de integracion numerica. Así tomando una triangulación de el valor de la integral es el siguiente. Una vez obtenida, la aproximacion de la integral queremos saber cuan buena es, para ello nos valemos del siguiete resultado. Podemos observar que el error cometido en la aproximacion depende principalmente del conjunto de nodos. Es conocido que se obtienen buenos resultados tomando un conjunto en el que los puntos están bien distribuidos. Por tanto, como hemos podido observar en la simulación anterior los puntos que obtenemos tienen una buena distribución luego darán lugar a una buena aproximación de esta integral.

Con esto concluye esta parte.

Competición en kaggle

En este trabajo se abordará la resolución de un problema planteado en la plataforma web Kaggle. Esta plataforma ofrece a sus usuarios la posibilidad de participar en distintas competiciones relacionadas con la Ciencia de Datos. Muchas de estas competiciones son problemas reales planteados por empresas, esto hace que el nivel de los participantes sea muy elevado ya que las primeras plazas de estos retos suelen estar recompensados con cuantiosas sumas de dinero o con la posibilidad de obtener un puesto de trabajo en dichas empresas.Debido a las limitaciones temporales y de preprocesamiento, el objetivo de este trabajo no es sólo obtener la mejor clasificación posible sino estudiar cómo de importante

es realizar un buen preprocesamiento y una buena elección del algoritmo a usar, frente a la capacidad de procesamiento de una máquina más potente.n nuestro caso, he participado en la competición TalkingData AdTracking Fraud Detection Challenge que estuvo activa desde el 5 de marzo de 2018 hasta el 7 de mayo de 2018. El problema fue planteado por la empresa china TalkingData, siendo su objetivo reducir el fraude en los anuncios de apps. Este fraude se produce cuando se registran clicks en los anuncios y estos no conllevan la instalación de la app. Por tanto, se produce una facturación de los canales publicitarios que no concuerda con la realidad. El desafío propuesto consiste en obtener un modelo para predecir si un usuario descargará una aplicación después de hacer clic en un anuncio de aplicación móvil. Para llevar a cabo esta tarea han proporcionado un conjunto de datos que cubre aproximadamente 200 millones de clics. Estos datos contienen los siguientes atributos: ip,os,device,channel,app,click_time,click_attribute,is_attributed.

El proceso a seguir será el siguiente: En primer lugar estudiaremos los datos proporcionados, de este modo podremos obtener algunas pistas sobre las operaciones a realizar durante la fase de preprocesamiento. Para ello, visualizaremos cómo se distribuyen los distintos datos, el número de valores vacíos de cada atributo y las posibles relaciones entre los distintos atributos. A continuación, probaremos distintas técnicas de preprocesamiento con el fin de obtener el mejor conjunto de datos posible.Finalmente, ejecutaremos distintos algoritmos para obtener el mejor clasificador posible.

Preprocesamiento

Para poder elegir una estrategia para el preprocesamiento es necesario realizar una visualización de los datos. De esta forma, podremos obtener cómo están distribuidos los valores de cada uno de los atributos o si existe alguna relación de correlación entre ellos.

Tras realizar el estudio de los datos se obtienen las siguientes conclusiones * La columna $attributed_time$ puede ser eliminada ya que la mayoría de sus valores son vacíos. * De la variable *click_time* podemos obviar los datos relativos al mes y al año. * Las variables *os* y *device* representan la misma información. * Existe un gran desbalanceo de clases.

Una vez hemos estudiado los datos que tenemos, vamos a realizar diferentes modificaciones en ellos en aras de obtener un mejor conjunto de entrenamiento. En primer lugar, como consecuencia directa del estudio anterior haremos los siguientes cambios. * Eliminar la columna attributed_time * Obtener los datos timestamp y día de la variable click_time A continuación, dado que las variables de las que disponemos son de tipo categórico y viendo la distribución que tienen se nos ocurre agrupar distintas combinaciones de ellas. A la hora de hacer estas agrupaciones he considerado 2 formas de hacerlo: contar el numero de valores repetidos y usar el valor medio. La siguiente tabla resume las pruebas

realizadas durante esta fase. Se puede ver que el conjunto de partida nos dio una puntuacion de 0.83 y que la mejora mas significativa se obtuvo al agrupar.

Algoritmos

Boosting (xgboost) como algoritmo inicial, para mejorar su rendimiento he optimizado los valores de algunos parámetros usando el concepto de grid search, esto es probar todas las combinaciones posibles. Obviamente este es un proceso costoso por lo que no es muy rentable, pero te asegura un mejor resultado porque te ajusta el algoritmo a tus datos.

Dado el desbalanceo de clases se opto por buscar algoritmos alternativos, en esta linea se encontro el algoritmo RUS y CUS. Ambos son una versión de boosting, se diferencian en que Rus y cus realizan undersampling sobre el conjunto de entrenamiento que se le pasa a cada clasificador. Rus realiza un random undersampling y Cus realiza undersampling con el metodo de los centroides.

Resultados obtenidos

- Los resultados obtenidos en esta última sección no han sido tan satisfactorios como a priori esperaba, ya que la aplicación de algoritmos alternativos no ha mejorado los resultados.
- Tras obtener los mejores parámetros del algoritmo Boosting hemos mejorado la puntuación un 2,73%.
- Durante la fase de preprocesamiento hemos mejorado el rendimiento de nuestro clasificador un 3 % aproximadamente.

Esto pone en relieve el porqué la mayoría de los participantes se decanta por el uso de xgboost, ya que al ser una librería de código abierto tiene un gran soporte que permite que el algoritmo esté muy optimizado. Sin embargo, los resultados obtenidos por RUSBoost parecen prometedores. Por otro lado, hemos constatado la importancia de la fase de preprocesamiento ya que hemos conseguido una gran mejora sin importar la capacidad de procesamiento de nuestras máquinas. Esto nos lleva a concluir que en situaciones donde la importancia de procesamiento no fuera tan relevante, habríamos obtenido una mejor clasificación final.