Catálogo de proyectos

Programa Experto en Big Data
2016-2017

[CON01] Business Intelligence sobre Big Data

Tutor: Julio Conca

Descripción: Los proyectos Big Data están englobados en dos grandes conjuntos, operacionales y de investigación. Los proyectos operacionales serían tipos de proyectos que ya se están realizando con las herramientas tradicionales pero utilizando las tecnologías Big Data para realizarlos sobre más información, más rápido y/o de forma más económica. Son los proyectos que están realizando las empresas que quieren aproximarse al mundo Big Data. Business Intelligence, la habilidad para transformar los datos en información, y la información en conocimiento, de forma que se pueda optimizar el proceso de toma de decisiones en los negocios, sería uno de esos proyectos de mejora operacional. En este trabajo partiríamos de datos meteorológicos para transformarlos en conocimiento y poder consumirlos de una forma visual.

Objetivo:
- Introducirse en uno de los casos de uso más comunes de introducción al Big Data para las empresas.
- Crear cuadros de mando con un enfoque más clásico (Herramientas de BI y HiveQL)
- Crear cuadros de mando con un enfoque más artesanal (D3 y HBase)

Tecnologías a usar: Flume, Map/Reduce, Hive, Impala HBase, D3 y Tableau o similares.

Origen de los datos: ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/

Plan básico a seguir:
- Exploración y subida a HDFS del dataset elegido.
- Definición de gráficos a realizar.
- Procesado de la información.
- Carga del resultado en Hive, Impala
- Implementación de los gráficos con Tableau o similares
- Carga del resultado en HBase.
- Implementación de los gráficos en D3.

[IVA01] Spark, Visión artificial con TensorFlow, e inferencia entre categorías

Tutor: Iván de Prado Alonso
Descripción: La introducción de modelos de redes neuronales profundas (deep learning) está revolucionando el mundo de la visión artificial. A su vez, las tecnologías como Spark están permitiendo el procesamiento masivo de datos. En este proyecto se propone usar el clasificador de imágenes ya preentrenado de TensorFlow Inception-V3 (https://www.tensorflow.org/tutorials/image_recognition) pero de forma que la inferencia se ejecute de forma paralelizada en Spark. La idea es clasificar en paralelo las imágenes del dataset de Flickr MIRFLICKR (http://press.liacs.nl/mirflickr/mirdownload.html) usando el modelo de Inception-V3, que clasifica en 1000 diferentes categorías (ver aquí: http://image-net.org/explore.php).

Por otro lado las imágenes del dataset de flicker vienen con los tags que la gente de la plataforma ha usado con las imágenes. La tarea final de este proyecto es tratar de encontrar las equivalencias entre las categorías de Image-net y los tags de Flickr, de forma que dado un tag de flicker se pueda indicar cuales son las categorías de Image-net más probables y viceversa.

Objetivo:
- Aprender a paralelizar una tarea usando Spark
- Familiarizarse con las nuevas técnicas de visión artificial, usando un modelo pre-entrenado
- Enfrentarse a un problema de análisis de datos como es el de inferir las relaciones entre dos categorizaciones, la de tagging y la inferida por el modelo
- de visión artificial.

Tecnologías a usar:
- Spark
- TensorFlow
- Inception-V3
- Otras que se consideren oportunas.

Origen de los datos:
http://press.liacs.nl/mirflickr/mirdownload.html

Plan básico a seguir:
1 - Ser capaz de hacer inferencia con TensorFlow Inception-V3 sobre las imágenes
2 - Desarrollar un sistema en Spark que permita paralelizar la inferencia de las imágenes
3 - Cruzar las inferencias obtenidas en las imágenes con los tags de Flickr para obtener un modelo que las relacione
4 - Elaboración de documentación y video demostrativo que incluya la ejecución en distribuido (más de una máquina, aunque sea virtual) del proyecto.

[IVA02] Análisis de sentimiento en reseñas de productos de Amazon usando Spark MLLib

Tutor: Iván de Prado Alonso

Descripción: La detección automática del sentimiento en reseñas o comentarios de usuarios en redes sociales permite a las empresas mejorar su marca: una empresa capaz de reaccionar ante comentarios negativos puede ponerlos solución de forma que su servicio y por tanto su imagen de marca, sea mejor. Así pues, el objetivo de este proyecto es tratar de clasificar reseñas/comentarios en positivos o negativos. Para ello, se hará uso del siguiente dataset de Amazon: https://snap.stanford.edu/data/web-FineFoods.html.

Este dataset contiene reseñas para muchos productos de Amazon, así como su valoración por parte del cliente, lo cual nos permite saber si el comentario fue positivo o negativo. Podemos usar estos datos, por tanto, para entrenar un modelo de "machine learning" que sea capaz de inferir para cualquier otro comentario futuro si es negativo o positivo. La siguiente página (https://www.kaggle.com/gpayen/d/snap/amazon-fine-food-reviews/building-a-prediction-model) contiene un ejemplo de como hacerlo usando Python y Scikit-learn. La idea sería implementar un modelo similar pero usando Spark MLLib, explorando posibles modelos como el mísmo de la página (Naïve Bayes) u otros como puedan ser "decision trees", "random forest", etc. Opcionalmente si se quiere ampliar el alcance del proyecto, se puede explorar la realización de sistemas de recomendación de productos, o la predicción del score en base al comentario, o la predicción de la relevancia del comentario en base al texto, todo ello usando el mismo conjunto de datos y la misma librería MMLib.

Objetivo:

- Aprender lo básico del "machine learning" usando unos clasificadores sencillos ya implementados
- Utilizar Spark como herramienta Big Data

Tecnologías a usar:

- Spark MLLib
- Otras que se consideren oportunas

Origen de los datos:

https://snap.stanford.edu/data/web-FineFoods.html

Plan básico a seguir:

1 - Familiarizarse con lo básico del "machine learning"
2 - Familiarizarse con las técnicas básicas del "text mining": stemmin, stop words removal, lowering, tokenization, pruning numbers and punctuation
3 - Implementación de la carga y preprocesado de los datos en Spark.
4 - Modelado usando Naïve Bayes
5 - Modelado con otros algoritmos: "decision trees", "random forest"
6 - Evaluación de los resultados
7 - Opcionales:

- Entrenar un sistema de recomendación
- Entrenar un predictor de la puntuación
- Entrenar un clasificador de la relevancia de los comentarios

8 - Elaboración de documentación y video demostrativo que incluya la ejecución en distribuido (más de una máquina, aunque sea virtual) del proyecto.

[MIG01] Desarrollo de un data federation basado en Spark

Tutor: Miguel Ángel Fernández Díaz

Descripción del trabajo: Apache Spark es un framework concebido para que el usuario final tenga que hacerse cargo del registro de tablas y configuración del cluster. Sin embargo, en muchas ocasiones el usuario final no tiene conocimientos ni de lenguajes de programación, ni de configuraciones de Spark. Es por ello, que un Data Federation abstrae de toda esta complejidad a este tipo de usuario ya que, una vez desplegado, configurado y registrado, este tipo de usuarios ya solo tendría que ocupar de realizar analítica a través de un lenguaje SQL. La idea de este proyecto es realizar una aplicación cliente servidor a través de Akka, donde el servidor sería un driver de Apache Spark y donde se realizaría el registro de las tablas en Apache Zookeeper para que fuera persistido y poder estar disponible para otros despliegues de la aplicación. Además, el cliente de la aplicación enviará texto con sentencias SQL al servidor de aplicación a través de Akka y estas sentencias serán ejecutadas a través de Spark SQL.

Objetivo: Crear una aplicación multiusuario basada en Spark

Tecnologías a usar: Apache Spark, Apache Zookeeper, Apache Hadoop (HDFS), Apache Cassandra, Akka, Scala

Origen de los datos: Datos reales de vuelos en Estados Unidos

Plan básico a seguir: desarrollo de un cliente-servidor básico de Akka, persistencia de metadatos en Apache Zookeeper, implementación del ejecutor de queries en Apache Spark, ingesta de datos en HDFS (formato CSV), ingesta de datos en Cassandra, despliegue de la aplicación con 2 instancias compartiendo los metadatos y analítica de datos mediante sentencias SQL.

[MIG02] Creación de cubos OLAP con Apache Kafka y Apache Spark

Tutor: Miguel Ángel Fernández Díaz

Descripción del trabajo: Uno de los procesos más costosos en los entornos de Big Data y de los entornos de analítica en general, es tener que realizar operaciones "full-scan" para la obtención de métricas y el análisis de datos. Es por ello que la generación de cubos OLAP, cuando es posible, para la agrupación de datos en tiempo real, y antes de ser persistidos, es una de las técnicas más usadas en entornos con data lakes. Así, la idea es crear una aplicación con Spark Streaming que reciba datos a través de Apache Kafka y, en base a una configuración, generar cubos OLAP y persistirlos en Apache Cassandra. Por otro lado, los datos "en bruto", serán persistidos en HDFS con formato Apache Parquet.

Objetivo: Aceleración de consultas a través del cálculo al vuelo de datos en streaming antes de ser persistidos

Tecnologías a usar: Apache Spark, Apache Kafka, Apache Hadoop (HDFS), Apache Parquet, Apache Cassandra, Scala

Origen de los datos: Simulador basado en datos reales de taxis de Nueva York

Plan básico a seguir: desarrollo de un generador de datos que envié datos a un topic de Kafka, desarrollo de una aplicación para la recepción de esos datos a través de Spark Streaming, desarrollo para la persistencia de esos datos "en bruto" en HDFS, desarrollo para la generación de cubos OLAP, comparación de rendimiento y tiempos entre analítica batch sobre datos "en bruto" y analítica batch sobre cubos OLAP.

[DAN01] Detección de anomalías en streaming

Tutor: Daniel Higuero

Descripción del trabajo: Dentro del mundo del IoT cada vez encontramos mas variedad de elementos sensorizados. El análisis de señales en streaming representa uno de los problemas a resolver en este tipo de entornos, y es por esto que este trabajo se centra en cómo plantear una solución de analítica en streaming que sea capaz de soportar un conjunto heterogéneo de elementos. Dentro los posibles análisis, este proyecto se centra en la detección de anomalías dentro de una señal monovariable. El alumno deberá decidir qué métodos de análisis utilizar para encontrar anomalías dentro de la señal y definir una arquitectura que permita su procesado incluyendo el desarrollo necesario para una prueba de concepto. Nótese que el reto propuesto no esta tan relacionado con la complejidad inherente de las señales a tratar o los algoritmos seleccionados como la ejecución de los mismos en streaming y la capacidad del sistema de procesar n señales en paralelo.

Objetivo: Analizar señales heterogéneas en streaming

Tecnologías a usar: A definir por el alumno

Origen de los datos: Se proporcionará una señal de referencia que será complementada con otras de las públicas disponibles.

Plan básico a seguir:
- Definición del método de detección de anomalías a utilizar
- Definición de la arquitectura propuestas y componentes principales.
- Selección de tecnologías a utilizar
- Implementación de una PoC del sistema

[JOR01] Monetización de datos usando Big Data

Tutor: Jorge López-Malla

Descripción del trabajo: Las tecnologías Big Data han demostrado, de sobra, su potencial como herramientas de procesamiento, pero ¿pueden aportar algo más a las empresas?. La capacidad de mezclar cantidades masivas de datos y poder aplicar algoritmos de Machine Learning de una manera sencilla nos abre un amplio abanico de posibilidades.

En este trabajo nos acercaremos a una posibilidad que es muy demandada en el mundo empresarial, la monetización de los datos. Para ello usaremos un dataset autogenerado de eventos telefónicos, red de antenas y clientes de una empresa telefónica para demostrar que los datos pueden monetizarse en sectores distintos al propio de la empresa.

Objetivo:
- Familiarizarse con algunas delas herramientas Big Data más usadas actualmente.
- Gestionar todas las fases de un proceso Big Data, desde la ingesta hasta la representación.

Tecnologías a usar:
- Spark
- HDFS
- Cassandra
- Herramientas de visualización de datos (Tableau o similares.)

Origen de los datos: Dataset autogenerado de eventos telefónicos.

Plan básico a seguir:
- Saneamiento e ingesta del dataset a HDFS en formato parquet.
- Procesamiento de las fuentes y realización del modelo usando KMeans.
- Análisis de los resultados del modelo.
- Agregación de usuarios por patrones.
- Ingesta de las agregaciones en un repositorio final distribuido apropiado
- Desarrollo de gráficos para la monetización de los datos

Otras fuentes de datos

Con el fin de facilitar la propuesta de otros trabajos por parte del alumno, a continuación se indican otras fuentes de datos que pueden ser utilizadas como dataset principal o complementario en un proyecto

- Datos relativos a los viajes en taxi en New York: http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml
- Datos relativos a viajes en avión: http://stat-computing.org/dataexpo/2009/
- Conjunto de datasets públicos de Amazon AWS: https://aws.amazon.com/datasets/
- Actividad en GitHub: https://www.githubarchive.org/
- Datasets públicos de Google: https://research.google.com/research-outreach.html#/research-outreach/research-datasets
- Yelp Challenge: https://www.yelp.com/dataset_challenge
- Planetlab logs: https://www.planet-lab.org/planetlablogs