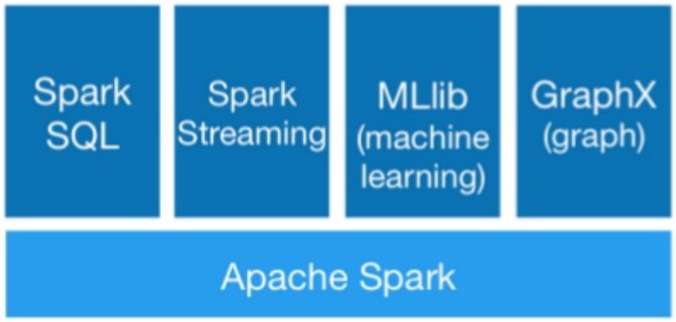
Note Taker:
Lily Kahler

Spark for ML and AI

Instructor:
Dan Sullivan

Apache spark is one of common tools used for big data. In this document, you learn about data preparation, preprocessing with Spark and post processing using ML algorithms. Spark support multiple languages such as Scala, Java, Python and etc. Spark has a modular system allowing multiple components; MLib for machine learning, SQL for relational querying, Streaming for continuous process of streaming data, and GraphX for graph analysis. Spark is used to manage large data set in real time, offline, text analysis, and etc.



The first step in Spark is preprocessing, which includes collecting, reformatting, and transforming data. The next stage is building a model using ML, and final step is validation step which asses the quality of model built in stage 2.

During preprocessing, we take care of missing or invalid data, normalizing the range value of features, standardizing categorical data (for example showing country names using 3 letter ISO codes)

In stage of building model, we select the algorithms, executing it on our data, and tuning the hyperparameters (like number of levels in decision tree).

In validation stage, model is applied on additional test sets, measuring the quality of models (accuracy, precision, and sensitivity_recall). In practice, we typically do these steps repeatedly in circle (preprocessing, model building, validation, preprocessing, validation, and so on). Each iteration gives new information about our assumption and data and help us to hone the model.

You may download spark directly from the website. We use pyspark which allows us to use python. Data frame in spark and python represent the same thing. The MLlib package has 3 types of functions: algorithms (classification…), workflows (pipeline, tuning hyperparamters…), and utilities (distributed math libraries).

Data Preparation and Transformation

There are 2 types of preprocessing, numeric and text preprocessing. **Normalizing (MinMaxScaler)** the data, maps values between 0 and 1. It avoid problem of large range of distribution. For example, salaries have high range and years of employments are in small value range. Normalizing the data helps avoiding giving much bigger weight to one feature based on the value in compare to other in the algorithm. The largest value maps to 1 and smallest data will map to 0 and whatever in between get values in between. **Standardizing (StandardScaler)** the data maps values between -1 and 1. It creates a data with mean value of 0 and standard distribution of 1. This is a bell shape formation known as normal distribution. We use standardization when the algorithm has normal distribution assumption such as SVM and some linear models. **Partitioning (Bucketizer)** data is used when we would like to work with group of data instead of continuous range of values. It maps data to buckets. Deciles and percentile are example of buckets. **Tokenizing** text maps single strings to the set of tokens. For example, "this is a sentence" tokenize into ["this", "is", "a", "sentence"]. **Term frequency inverse document frequency (TF-IDF- transformation)** maps string to a vector indicating the frequency of each word in a text relative to the group of texts (HashingTF). It is widely used in classification. Basically, it tokenizes the sentences and then counts the number which each work appears in the corpuse. Corpuse is a collection of text documents. We count how often a work appears in a document and how often it appears in corpus. Then we feed these two parameters into TF-IDF calculation. It gives us the TF-IDF measures.

Clustering (unsupervised ML)

Sometimes we need to give macro level structure to the data using clustering. K-means is mainly used to find clusters in small and midsized dataset. For larger dataset, hierarchical clustering with Bisecting K-means. It is easily implemented in pyspark as shown in example session on my website.

Classification (supervised ML)

We use it for classifying the data into different categories. Classification algorithms help us identifying boundaries and make it easy to assign new entities into the category. Most commonly used algorithms are Naive Bayes (good for independent variables), decision tree (good in many cases), multilayer perceptron (type of neural network, goof for nonlinear relationship between variables). The example file can be found on the main page of website for this section.

Regression (supervised ML)

Regression allow us to make prediction over numerical values. In the main section of website, you can find the code for linear regression, decision tree regression and gradient boosted tree regression.

Few tips

During preprocessing using Spark, it's good practice to:

- Load data into dataframe
- Include headers or column names
- Use inferSchema=True, this helps the data get mapped to the appropriate data type
- Use VectroAssembler to create feature vectors
- Use stringIndexer to map from string to numeric indexes

During the model building phase:

- Split data randomly into training and test set
- Fit model in training data
- Create prediction by applying transform to the test data

During validation:

- Use the right MLlib evaluator (MultiClassificationEvaluator, RegressionEvaluator…)
- Experiment with multiple models
- Vary hyperparameters