Spark Machine Learning Overview
Spark Machine Learning API

- Original `spark.mllib` API
  - Primary API using RDDs
- "Pipelines" `spark.ml` API
  - Experimental higher-level API for constructing ML workflows using DataFrames
Example: Log prioritization

Data: Test logs

Goal: Prioritize logs to investigate

Running test:
pyspark/conf.py
Spark assembly has been
built with Hive,
including Datanucleus
jars on classpath
14/12/15 18:36:12 WARN
Utils: Your hostname,

Running test:
pyspark/broadcast.py
Spark assembly has been
built with Hive,
including Datanucleus
jars on classpath
14/12/15 18:36:30 ERROR
Aliens attacked the

-1.1
1.9
Example: Log prioritization

**Instance**

Running test:
pyspark/conf.py
Spark assembly has been built with Hive, including Datanucleus jars on classpath
14/12/15 18:36:12 WARN
Utils: Your hostname,

**Label**

-1.1

How can we learn?
- Choose a model
- Get training data
- Run a learning algorithm
A model is a function \( f : x \rightarrow y \)

\( \text{Instance} \quad x \quad \rightarrow \quad \text{Label} \quad y \)

Running test: pyspark/conf.py
Spark assembly has been built with Hive, including Datanucleus jars on classpath
14/12/15 18:36:12 WARN Utils: Your hostname,

Convert to features, e.g., word counts

\( \text{Running} \quad \text{test} \quad \text{Spark} \quad \text{aliens} \)
\( 43 \quad 67 \quad 110 \quad 0 \quad \bullet \quad \bullet \quad \bullet \bullet \)

\( \text{Feature vector} \ x \quad \text{mllib.linalg.Vector} \)
A model is a function \( f: x \rightarrow y \)

**LinearRegression**

Our model: Parameter vector \( \mathbf{w} \)

<table>
<thead>
<tr>
<th>( w )</th>
<th>( 0.0 )</th>
<th>( 0.1 )</th>
<th>( -0.1 )</th>
<th>( 2.1 )</th>
<th>( \cdot \cdot\cdot )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w^T x )</td>
<td>( 0.0 )</td>
<td>( +6.7 )</td>
<td>( +(-11.0) )</td>
<td>( +0.0 )</td>
<td>( \cdot\cdot\cdot= -1.1 )</td>
</tr>
</tbody>
</table>

\( \uparrow \) Running \( \uparrow \) test \( \uparrow \) Spark \( \uparrow \) aliens

\( \downarrow \) \( \downarrow \) \( \downarrow \) \( \downarrow \)
Data for learning

```
Running test: pyspark/conf.py
Spark assembly has been built with Hive, including Datanucleus jars on classpath
14/12/15 18:36:12 WARN Utils: Your hostname,
```

LabeledPoint(features: Vector, label: Double)
Data for learning

<table>
<thead>
<tr>
<th>Instances</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running test: pyspark/conf.py Spark assembly has been built with Hive, including Datanucleus jars on classpath 14/12/15 18:36:12 WARN Utils: Your hostname,</td>
<td>-1.1</td>
</tr>
<tr>
<td>Running test: pyspark/conf.py Spark assembly has been built with Hive, including Datanucleus jars on classpath 14/12/15 18:36:12 WARN Utils: Your hostname,</td>
<td>2.3</td>
</tr>
<tr>
<td>Running test: pyspark/conf.py Spark assembly has been built with Hive, including Datanucleus jars on classpath 14/12/15 18:36:12 WARN Utils: Your hostname,</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Dataset

RDD[LabeledPoint]
ML algorithms

Recall:

A model is a function: features → label

\[
\text{LinearRegressionModel.predict(features: Vector): Double}
\]

A training dataset is a set of (features, label) pairs

\[
\text{RDD[LabeledPoint]}
\]

An ML algorithm is a function: dataset → model

\[
\text{LinearRegression.train(data: RDD[LabeledPoint]): LinearRegressionModel}
\]
Workflow: training + testing

Dataframe → Estimator → ML model

New logs

Transformer → ML model → Predicted priorities

Training

Testing

New API (“Pipelines”)
+ Evaluation

Training
- Training data
  - Estimator
    - Model (Transformer)

Testing
- Test data
  - Transformer
    - Predicted priorities

Evaluation
- Predicted labels + true labels (on test data)
  - RDD[(Double, Double)]
  - Evaluator
    - Regression
      - Metrics
        - Metric (MSE)
          - Double

Model selection
- Choose among different models or model hyperparameters

New API ("Pipelines")
Summary: ML Overview

Components
- Model
- Dataset
- Algorithm

Processes
- Training – Test – Evaluation
- Model selection
Overview of ML Algorithms

• Prediction
  • Regression
  • Classification
• Feature transformation
• Recommendation
• Clustering
• Other
  • Statistics
  • Linear algebra
  • Optimization
Overview of ML Algorithms

• Prediction
  • Regression
  • Classification

• Feature transformation

• Recommendation

• Clustering

• Other
  • Statistics
  • Linear algebra
  • Optimization

E.g.: given log file, predict priority

\textbf{LinearRegression, DecisionTree, LogisticRegression, NaiveBayes, SVM}

Iterative optimization

Partition data by rows (instances):
• Easy to handle billions of rows
• Hard to scale # features
  • \(10^7\) for *Regression, SVM, NB
  • \(10^3\) for DecisionTree
Overview of ML Algorithms

- Prediction
  - Regression
  - Classification
- Feature transformation
- Recommendation
- Clustering
- Other
  - Statistics
  - Linear algebra
  - Optimization

E.g.: convert text to feature vectors

**Tokenizer, HashingTF, IDF, Word2Vec**

**Normalizer, StandardScaler**

Arguably the *most important* part of machine learning

Per-row transformation
Overview of ML Algorithms

- Prediction
  - Regression
  - Classification
- Feature transformation
- Recommendation
  - user → recommended products
- Clustering
- Other
  - Statistics
  - Linear algebra
  - Optimization

E.g.: Recommend movies to users

**ALS**

Phrased as matrix factorization
- Given (users x products) matrix with many missing entries,
- Find low-rank factorization.
- Fill in missing entries.
- Partition data by both users and products.
- Scales to millions of users and products.
Overview of ML Algorithms

• Prediction
  • Regression
  • Classification
• Feature transformation
• Recommendation
• Clustering
  feature vectors \(\rightarrow\) clusters (no labels)
• Other
  • Statistics
  • Linear algebra
  • Optimization

E.g.: Given news articles, automatically group articles by topics

KMeans, GaussianMixtureEM, (LDA)

Iterative optimization.
Local optima.
Partition data by rows.
• Easy to handle billions of rows
• Hard to scale # features
  • But that’s OK (statistics)
Overview of ML Algorithms

• Prediction
  - Regression
  - Classification

• Feature transformation

• Recommendation

• Clustering

• Other
  - Statistics
  - Linear algebra
  - Optimization

E.g.: Is model A significantly better than model B?
Correlation, ChiSqTest, Statistics, MultivariateOnlineSummarizer

E.g.: Matrix decomposition
RowMatrix, EigenValue Decomposition, Matrix, Vector, ...

E.g.: Given function f(x), find x to minimize f(x)
GradientDescent, LBFGS
What’s coming?

- Infrastructure
  - ML Pipelines (graduate from experimental/developer status)
  - Model import/export
- Algorithms
  - A-priori & fp-growth (market basket analysis)
What’s missing?

• Time series
• Graphical models
  • Deep learning
• Anomaly detection
• Application areas
  • Natural Language Processing
  • Vision
  • Signal processing
ML Questions?