CLOUD COMPUTING APPLICATIONS

Spark Naïve Bayes

Roy Campbell & Reza Farivar

Definition

Naive Bayes is a simple multiclass classification algorithm with the assumption of independence between every pair of features.

NaiveBayes implements multinomial naive Bayes.

Input is an RDD of LabeledPoint and an optionally smoothing parameter lambda

Output is a NaiveBayesModel

```
from pyspark.mllib.classification import NaiveBayes, NaiveBayesModel
from pyspark.mllib.linalg import Vectors
from pyspark.mllib.regression import LabeledPoint

def parseLine(line):
    parts = line.split(',')
    label = float(parts[0])
    features = Vectors.dense([float(x) for x in parts[1].split(' ')])
    return LabeledPoint(label, features)

data = sc.textFile('data/mllib/sample_naive_bayes_data.txt').map(parseLine)

# Split data aproximately into training (60%) and test (40%)
training, test = data.randomSplit([0.6, 0.4], seed = 0)

# Train a naive Bayes model.
model = NaiveBayes.train(training, 1.0)

# Make prediction and test accuracy.
predictionAndLabel = test.map(lambda p : (model.predict(p.features), p.label))
accuracy = 1.0 * predictionAndLabel.filter(lambda (x, v): x == v).count() / test.count()

# Save and load model
model.save(sc, "myModelPath")
sameModel = NaiveBayesModel.load(sc, "myModelPath")
```