Microsoft

Vowpal Wabbit on Spark

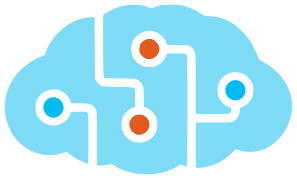
December 2019
Azure Global – Customer Engineering AI

Markus Cozowicz, Scott Graham, Wu Tao

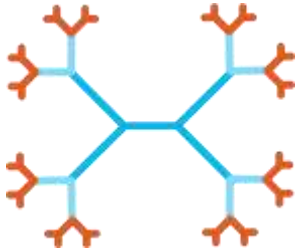
Chenhui Hu, Mark Hamilton, Ilya Matiach

Microsoft Machine Learning for Apache Spark

Vowpal Wabbit



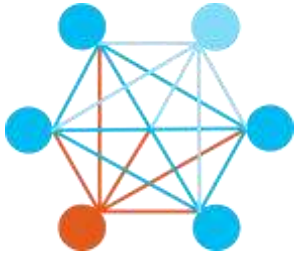
Cognitive Service



LightGBM

Spark Serving

HTTP on Spark



CNTK

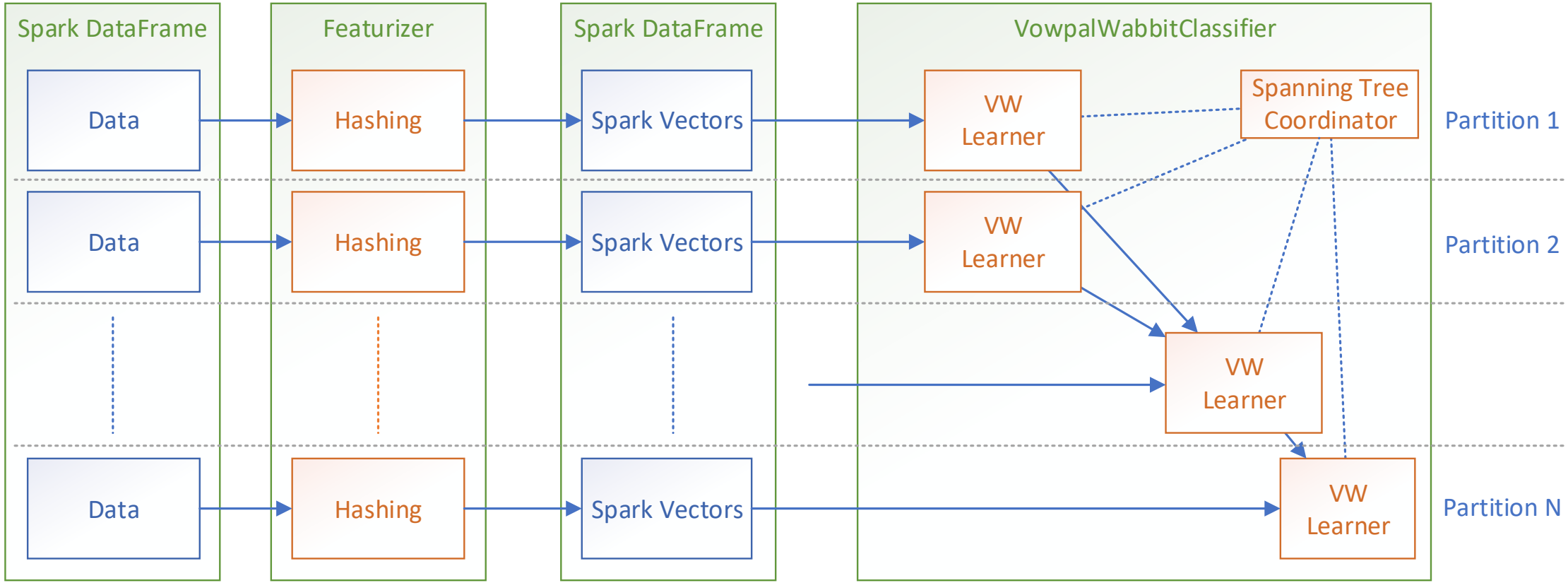
Lime

Binding Autogeneration

Vowpal Wabbit on Spark | Goals

Easy to use	→	Seamless integration w/ Spark ML
Blazing fast	→	Linear out-of-core learning
Distributed	→	Data parallel w/o manual cluster configuration
Scalable	→	10TB experiments

Vowpal Wabbit on Spark | Architecture



Microsoft

VOWPAL WABBIT

APACHE Spark

Vowpal Wabbit on PySpark | Featurize

```
from mmlspark.vw import VowpalWabbitFeaturizer

featurizer = VowpalWabbitFeaturizer(
    inputCols=['sepal length', 'sepal width', 'petal length'],
    outputCol='features')

df_train = featurizer.transform(df_input)
```

Vowpal Wabbit on Spark | Feature Engineering

Type	Sample Data	Vowpal Wabbit Format	Feature Inde[x|ices]	Feature Value(s)
Numeric	3.2	|col1 col1:3.2	hash(col1)	3.2
Boolean	True	|col1 col1:1	hash(col1)	1
String	NewYork	|col1 NewYork:1	hash(New York)	1
Array of String	[“Hello”,”world”]	|col1 Hello:1 World:1	hash(Hello), hash(World)	1, 1
Map of String to String	{“Income”:”High”, “Age”:”Middle”}	|col1 IncomeHigh:1 AgeMiddle:1	hash(IncomeHigh) hash(AgeMiddle)	1, 1
Map of String to Numeric	{“Speed”:12.3, “Velocity”:3}	|col1 Speed:12.3 Velocity:3	hash(Speed) hash(Velocity)	12.3, 3

- 1st namespace character = feature group used for interactions, n-grams, …
- Numeric: Double, Float, Integer, Long, Short, Byte

Microsoft

VOWPAL WABBIT

APACHE Spark

Vowpal Wabbit on PySpark | Train & Predict

```
from mmlspark.vw import VowpalWabbitClassifier

estimator = VowpalWabbitClassifier(
      args="--holdout_off --loss_function logistic --progress 1000000",
      learningRate=1.0,
      powerT=0.0,
      l1=1e-8)

model = estimator.fit(df_train)

predictions = model.transform(df_test)
```

Vowpal Wabbit on Spark | Hyper-parameter sweeps

Spark ML integration

Frequent parameters

- learningRate, powerT, l1, l2, interactions

Parallel cross-validation

```
val vw = new VowpalWabbitClassifier()
    .setArgs("--link=logistic")

val paramGrid = new ParamGridBuilder()
   .addGrid(vw.learningRate, Array(0.5, 0.05, 0.001))
   .addGrid(vw.numPasses, Array(1, 3, 5))
   .build()

val cv = new CrossValidator()
  .setEstimator(vw)
  .setEvaluator(new BinaryClassificationEvaluator)
  .setEstimatorParamMaps(paramGrid)
  .setNumFolds(3)
  .setParallelism(2)

val model = cv.fit(dataset)
model.transform(dataset)

val auc = model.avgMetrics(0)
```

Vowpal Wabbit on Spark | Bandit Support

Coming soon…

Multi-line examples

- Actions
- Multi-class

Label support

- action:cost:probability

Nested structures

Thanks!

VOWPAL WABBIT

Vowpal Wabbit on Spark | Features

Distributed AllReduce training

- Data partitions → compute jobs
- Limited by number of available executors

Multi-pass using VW cache file on disk

Performance metrics exposed as data frame

- average loss, total number of features,…

Native Model Export

Vowpal Wabbit on Spark | Features

Machine learning tasks

- Binary classification
- Regression

Data input

- Sparse and dense features using Spark Vector
- Example weights
- Namespace support by using multiple columns

VOWPAL WABBIT

Microsoft Research sponsored

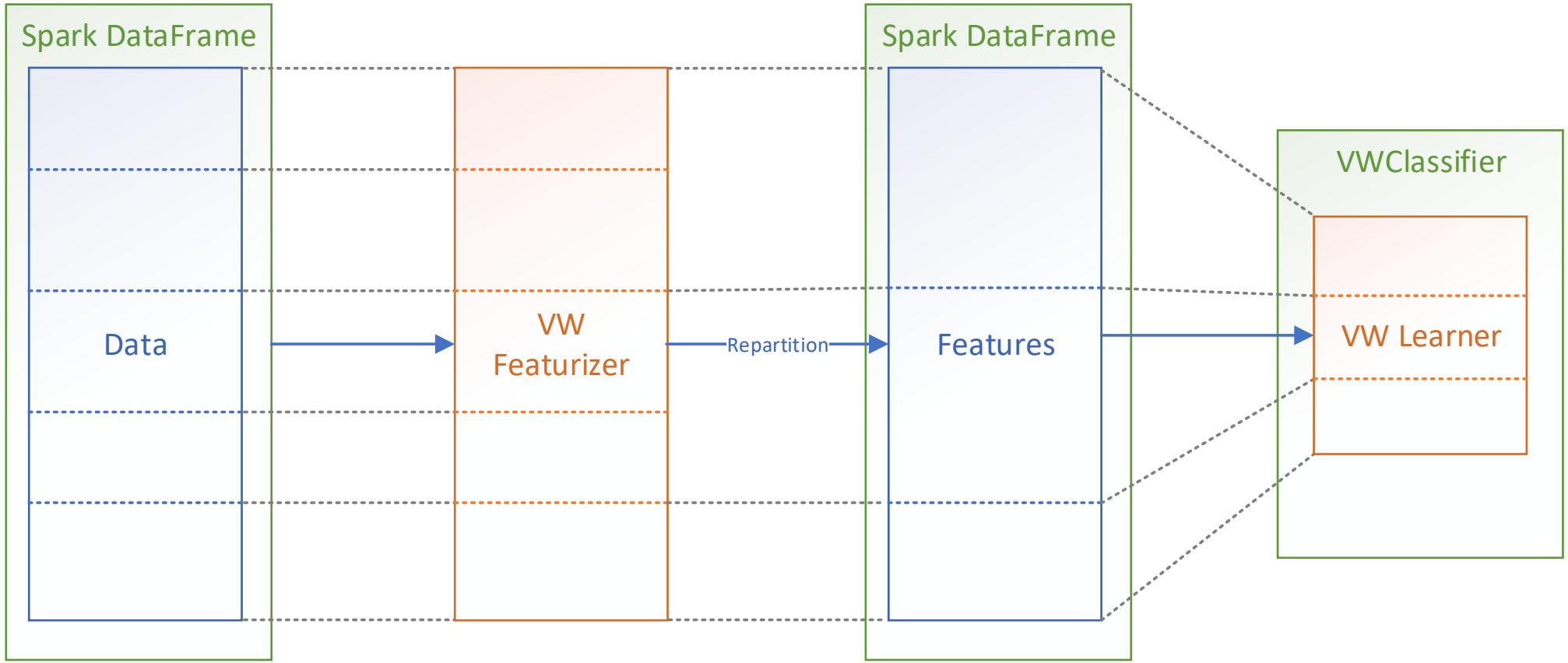
Suite of algorithms

- Supervised
- Unsupervised
- Reinforcement learning
  - Azure Cognitive Service Personalizer

Strengths

- Speed
- Memory efficiency
- Sparse features

Vowpal Wabbit on Spark | Optimizations



Repartitioning featurized data can reduce parallel training jobs

Vowpal Wabbit on Spark | Feature Engineering

```
new VowpalWabbitFeaturizer()
  .setInputCols(Array("title", "body"))
  .setOutputCol("features")
  .transform(df)
```

Option	Description
SumCollisions	Summed if true, otherwise the first is kept.
Seed	Seed used for hashing.
StringSplitInputCols	String column values split on whitespace.
PreserveOrderNumBits	Bits used to preserve order. If >0 inserts position at the highest order bits.
PrefixStringsWithColumnName	Prefixes string features with column name (default: true).

Vowpal Wabbit on Spark | N-Grams

Vowpal Wabbit supports n-gram generation at learning time

- Reduces memory footprint significantly

Obstacle

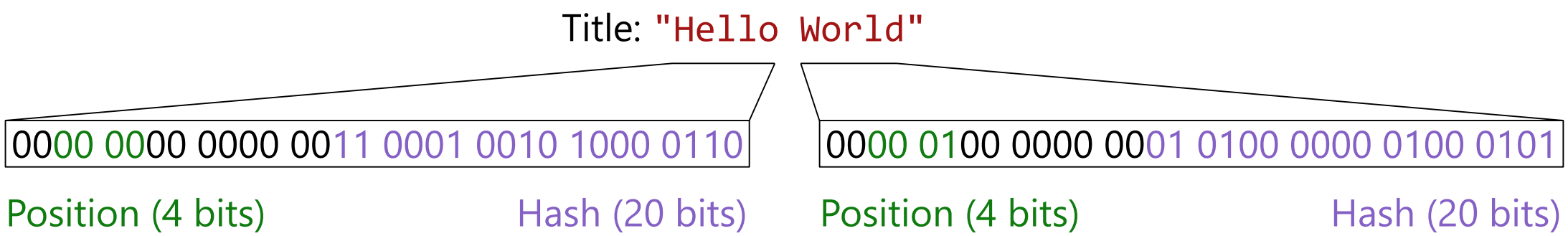
- Features must be presented in order of occurrence (e.g. text) for n-gramming
- Spark Sparse Vectors require **feature indices** to be sorted
- After hashing input text order is lost (indices = hash of word)

Solution

- Create composite feature indices (position + hash)
- Strip position for learning

Vowpal Wabbit on Spark | N-Grams

```
new VowpalWabbitFeaturizer()
   .setStringSplitInputCols(Array("title", "body"))
   .setPreserveOrderNumBits(4)
   .setNumBits(20)
   .setPrefixStringsWithColumnName(false)
   .transform(df)
```



Microsoft

VOWPAL WABBIT

APACHE Spark