Technische Universität Berlin

C10. Bonferroni Principle

- A data mining risk: discovering patterns that are meaningless
- Be careful not to draw the wrong conclusion
- Statistician's call it "Bonferroni's Principle"
  - If you look in more places for interesting patterns than your amount of data will support, you are bound to find crap.
- Example
  - TIA (Total Information Awareness) / Terrorism

Technische Universität Berlin

Interpretation

- Bonferroni's principle helps us avoid treating random occurrences (statistical flukes) as if they were real.
- Calculate the expected number of occurrences of the events you are looking for, on the assumption that data is random.
- If this number is **significantly larger** than the number of real instances you hope to find, then you must expect almost anything you find to be bogus, i.e., a statistical artifact rather than evidence of what you are looking for.
- This is an informal statement of Bonferroni's principle.

Technische Universität Berlin

The "TIA" Story

- Suppose we believe that certain groups of evil-doers are meeting occasionally in hotels to plot doing evil.
- We want to find (unrelated) people who at least twice have stayed at the same hotel on the same day.

Technische Universität Berlin

The Details

- 10^9 people being tracked.
- 1000 days.
- Each person stays in a hotel 1% of the time (10 days out of 1000).
- Hotels hold 100 people (so 10^5 hotels).
- If everyone behaves randomly (I.e., no evil-doers) will the data mining detect anything suspicious?

enough to hold the 1% of a billion people who visit a hotel on any given day.

Technische Universität Berlin

Calculations – (1)

p at some hotel

q at some hotel

Same hotel

- Probability that given persons *p* and *q* will be at the same hotel on given day *d*:
  - $\boxed{1/100} \times \boxed{1/100} \times \boxed{10^{-5}} = 10^{-9}$.
- Probability that *p* and *q* will be at the same hotel on given days d_1 and d_2:
  - $10^{-9} \times 10^{-9} = 10^{-18}$.
- Pairs of days:
  - 5×10^5.

Technische Universität Berlin

Calculations – (2)

- Probability that *p* and *q* will be at the same hotel on some two days:
  - $5\times10^{5} \times 10^{-18} = 5\times10^{-13}$.
- Pairs of people:
  - 5×10^{17}.
- Expected number of "suspicious" pairs of people:
  - $5\times10^{17} \times 5\times10^{-13} = 250{,}000$.

Technische Universität Berlin

Conclusion

- Suppose there are (say) 10 pairs of evil-doers who definitely stayed at the same hotel twice.
- Analysts have to sift through 250,010 candidates to find the 10 real cases.
  - Not gonna happen.
  - But how can we improve the scheme?

Technische Universität Berlin

Moral

- When looking for a property (e.g., "two people stayed at the same hotel twice"), make sure that the property does not allow so many possibilities that random data will surely produce facts "of interest."

Technische Universität Berlin

6. Data Analytics Libraries / Systems

Technische Universität Berlin

Representative Examples

1. Apache Singa
2. Spark MLlib
3. Apache Mahout
4. H_2O
5. ScalaNLP (Breeze)
6. GraphLab Create
7. MADLib
8. Apache Flink ML
9. Amazon Machine Learning
10. Microsoft Azure Machine Learning

Apache SINGA

A General Distributed Deep Learning Platform

Apache incubator

Introduction

Overview

SINGA is designed to be general to implement the distributed training algorithms of existing systems. Distributed deep learning training is an on- going challenge research problem in terms of scalability. There is no established scalable distributed training algorithm. Different algorithms are used by existing systems, e.g. Hogwild used by Caffe, AllReduce used by Baidu's DeepImage, and the Downpour algorithm proposed by Google Brain and used at Microsoft Adam. SINGA provides users the chance to select the one that is most scalable for their model and data.

To provide good usability, SINGA provides a simple programming model based on the layer structure that is common in deep learning models. Users override the base layer class to implement their own layer logics for feature transformation. A model is constructed by configuring each layer and their connections like Caffe. SINGA takes care of the data and model partitioning, and makes the underlying distributed communication (almost) transparent to users. A set of built-in layers and example models are provided.

SINGA is an Apache incubator project, released under Apache License 2. It is mainly developed by the DBSystem group of National University of Singapore. A diverse community is being constructed to welcome open-source contribution.

Goals and Principles

Goals

- Scalability: A distributed platform that can scale to a large model and training dataset.
- Usability: To provide abstraction and easy to use interface so that users can implement their deep learning model/algorithm without much awareness of the underlying distributed platform.
- Extensibility: to make SINGA extensible for implementing different consistency models, training algorithms and deep learning models.

Principles

Scalability is a challenge research problem for distributed deep learning training. SINGA provides a general architecture to exploit the scalability of different training algorithms. Different parallelism approaches are also supported:

- Model Partition—one model replica spreads across multiple machines to handle large models, which have too many parameters to be kept in the memory of a single machine. Overhead: synchronize layer data across machines within one model replica Partition.
- Data Partition—one model replica trains against a partition of the whole training dataset. This approach can handle large training dataset. Overhead: synchronize parameters among model replicas.
- Hybrid Partition—exploit a cost model to find optimal model and data partitions which would reduce both overheads.

To achieve the usability goal, we propose our programming model with the following two major considerations:

- Extract common data structures and operations for deep learning training algorithms, i.e., Back Propagation and Contrastive Divergence. Users implement their models by inheriting these data structures and overriding the operations.
- Make model partition and data partition automatically almost transparent to users.

Considering extensibility, we make our core data structures (e.g., Layer) and operations general enough for programmers to override.

Spark MLlib

spark.apache.org/mllib

Download | Libraries | Documentation | Examples | Community | FAQ

MLlib is Apache Spark's scalable machine learning library.

Ease of Use

Usable in Java, Scala, Python, and SparkR.

MLlib fits into Spark's APIs and interoperates with NumPy in Python (starting in Spark 0.9). You can use any Hadoop data source (e.g. HDFS, HBase, or local files), making it easy to plug into Hadoop workflows.

Performance

High-quality algorithms, 100x faster than MapReduce.

Spark excels at iterative computation, enabling MLlib to run fast. At the same time, we care about algorithmic performance: MLlib contains high-quality algorithms that leverage iteration, and can yield better results than the one-pass approximations sometimes used on MapReduce.

Easy to Deploy

Runs on existing Hadoop clusters and data.

If you have a Hadoop 2 cluster, you can run Spark and MLlib without any pre-installation. Otherwise, Spark is easy to run standalone or on EC2 or Mesos. You can read from HDFS, HBase, or any Hadoop data source.

Algorithms

MLlib contains the following algorithms and utilities:

- logistic regression and linear support vector machine (SVM)
- classification and regression tree
- random forest and gradient-boosted trees
- recommendation via alternating least squares (ALS)
- clustering via k-means, Gaussian mixtures (GMM), and power iteration clustering
- topic modeling via latent Dirichlet allocation (LDA)
- singular value decomposition (SVD) and QR decomposition
- principal component analysis (PCA)
- linear regression with L_1, L_2, and elastic-net regularization
- isotonic regression
- multinomial/binomial naive Bayes
- frequent itemset mining via FP-growth and association rules
- sequential pattern mining via PrefixSpan
- summary statistics and hypothesis testing
- feature transformations
- model evaluation and hyper-parameter tuning

Refer to the MLlib guide for usage examples.

Latest News

Submission is open for Spark Summit East 2016 (Oct 14, 2015)

Spark 1.5.1 released (Oct 02, 2015)

Spark 1.5.0 released (Sep 09, 2015)

Spark Summit Europe agenda posted (Sep 07, 2015)

Archive

Download Spark

Built-in Libraries:

SQL and DataFrames
Spark Streaming
MLlib (machine learning)
GraphX (graph)

Third-Party Packages

General Developers Mahout-Samsara Algorithms MapReduce Basics Mahout MapReduce

What is Apache Mahout?

The Apache Mahout™ project's goal is to build an environment for quickly creating scalable performant machine learning applications.

mahout.apache.org

The three major components of Mahout are an environment for building scalable algorithms, many new Scala + Spark (H2O in progress) algorithms, and Mahout's mature Hadoop MapReduce algorithms.

download mahout

07 Aug 2015 - Apache Mahout 0.11.0 released

Apache Mahout introduces a new math environment we call Samsara, for its theme of universal renewal. It reflects a fundamental rethinking of how scalable machine learning algorithms are built and customized. Mahout-Samsara is here to help people create their own math while providing some off-the-shelf algorithm implementations. At its core are general linear algebra and statistical operations along with the data structures to support them. You can use is as a library or customize it in Scala with Mahout-specific extensions that look something like R. Mahout-Samsara comes with an interactive shell that runs distributed operations on a Spark cluster. This make prototyping or task submission much easier and allows users to customize algorithms with a whole new degree of freedom.

Mahout Algorithms include many new implementations built for speed on Mahout-Samsara. They run on Spark 1.3+ and some on H2O, which means as much as a 10x speed increase. You'll find robust matrix decomposition algorithms as well as a Naive Bayes classifier and collaborative filtering. The new spark-itemsimilarity enables the next generation of cooccurrence recommenders that can use entire user click streams and context in making recommendations.

Visit our release notes page for details. Interested in helping? Join the Mailing lists.

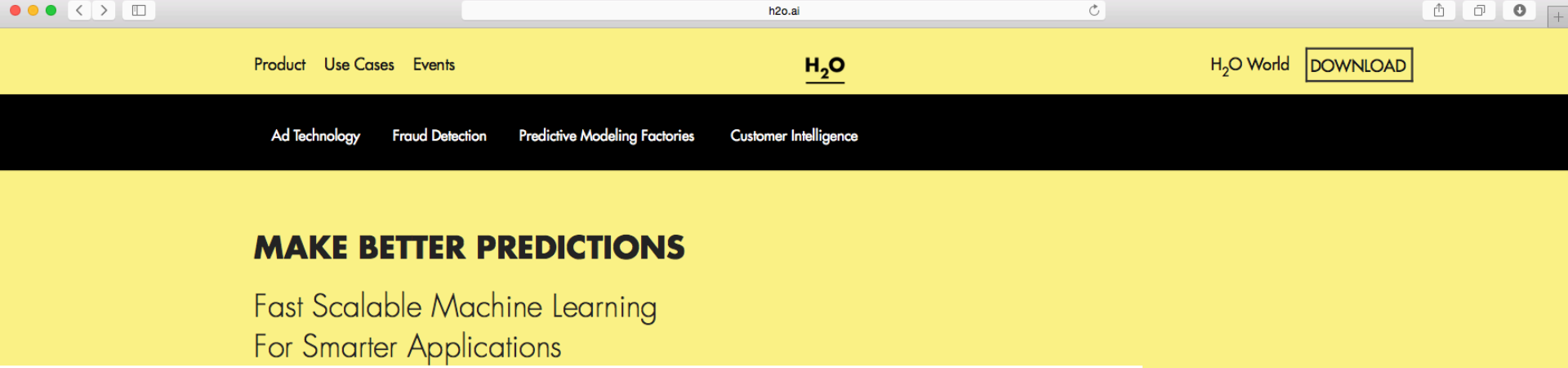
Latest release version 0.11.0 has

Mahout Samsara Environment

- Distributed Algebraic optimizer
- R-Like DSL Scala API
- Linear algebra operations
- Ops are extensions to Scala
- IScala REPL based interactive shell
- Integrates with compatible libraries like MLLib
- Run on distributed Spark
- H2O in progress

Mahout Samsara based Algorithms

- Stochastic Singular Value Decomposition (ssvd, dssvd)
- Stochastic Principal Component Analysis (spca, dspca)
- Distributed Cholesky QR (thinQR)
- Distributed regularized Alternating Least Squares (dals)
- Collaborative Filtering: Item and Row Similarity
- Naive Bayes Classification
- Distributed and in-core



Model with State of the Art Machine Learning Algorithms

Model	Description
Generalized Linear Models (GLM)	A flexible generalization of ordinary linear regression for response variables that have error distribution models other than a normal distribution. GLM unifies various other statistical models, including linear, logistic, Poisson, and more.
Decision Trees	A decision support tool that uses a tree-like graph or model of decisions and their possible consequences.
Gradient Boosting (GBM)	A method to produce a prediction model in the form of an ensemble of weak prediction models. It builds the model in a stage-wise fashion and is generalized by allowing an arbitrary differentiable loss function. It is one of the most powerful methods available today.
K-Means	A method to uncover groups or clusters of data points often used for segmentation. It clusters observations into k certain points with the nearest mean.
Anomaly Detection	Identify the outliers in your data by invoking a powerful pattern recognition model.
Deep Learning	Model high-level abstractions in data by using non-linear transformations in a layer-by-layer method. Deep learning is an example of unsupervised learning and can make use of unlabeled data that other algorithms cannot.
Naïve Bayes	A probabilistic classifier that assumes the value of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. It is often used in text categorization.
Grid Search	Is the standard way of performing hyper parameter optimization to make model configuration easier. It is measured by cross-validation of an independent data set.

1. H_2O Predictive Analytics
h2o.ai

2. Sparkling Water (H_2O in Spark)
github.com/h2oai/sparkling-water

www.scalanlp.org

scalanlp.org

ScalaNLP

Scientific Computing, Machine Learning, and Natural Language Processing

Home | About | People | Documentation | Models | Support | Blog

ScalaNLP is a suite of machine learning and numerical computing libraries.

ScalaNLP is the umbrella project for several libraries, including Breeze and Epic. **Breeze** is a set of libraries for machine learning and numerical computing. **Epic** is a high-performance statistical parser and structured prediction library.

Get Started with Breeze!

Breeze

Breeze is the core set of libraries for ScalaNLP, including linear algebra, numerical computing and optimization. It enables a generic, powerful yet still efficient approach to machine learning.

Epic

Epic is a powerful, state-of-the-art, statistical parser for eight languages backed by a generic framework for building complex systems using structured prediction.

Puck

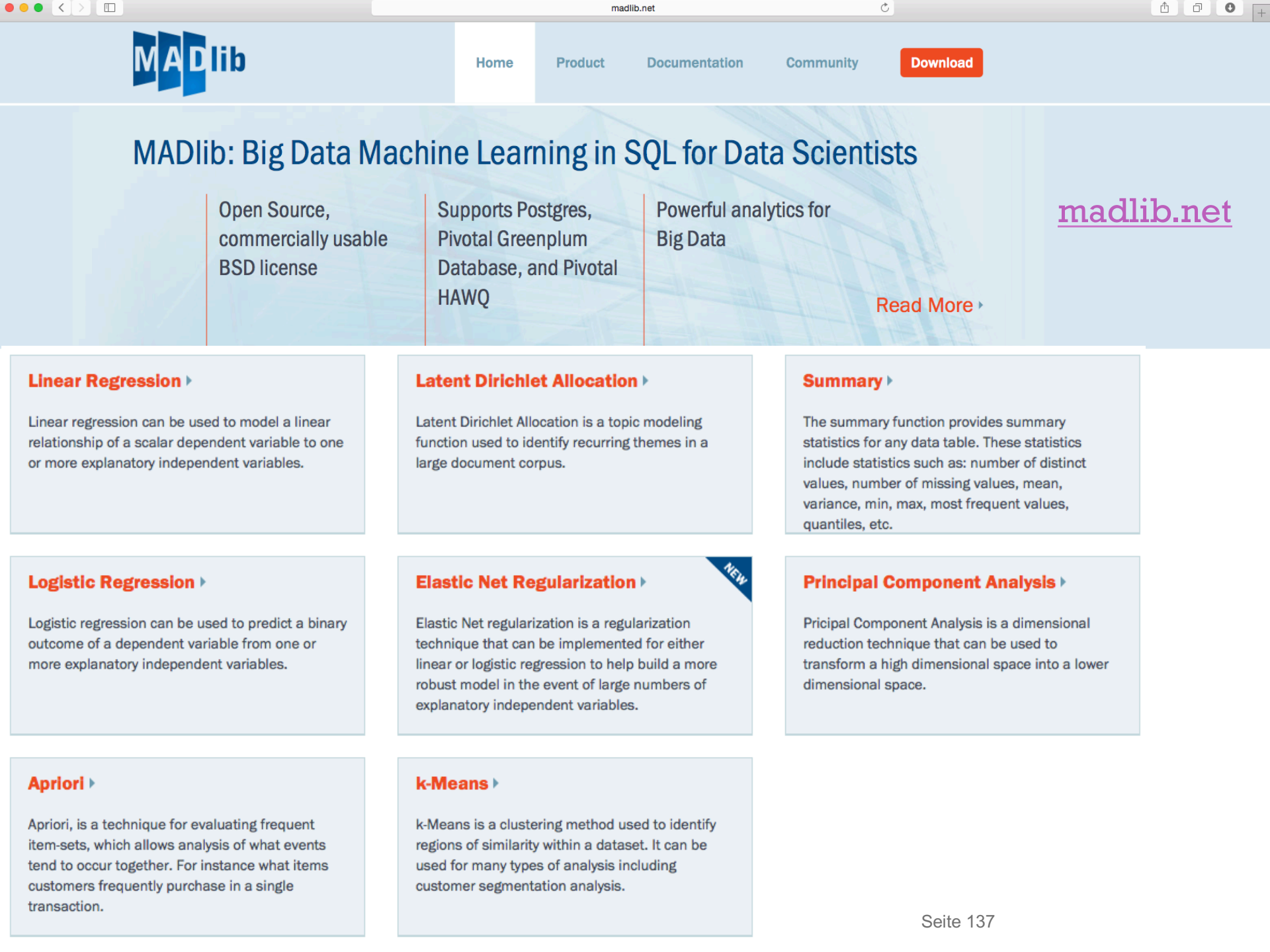
Puck is an insanely fast GPU-powered parser, built on the same grammars produced by the Berkeley Parser. On a mid-range Nvidia GTX 680, it can parse over 400 sentences a second, or over half a million words per minute.

GraphLab Create™ offers a deep library of Python machine learning APIs and toolkits

We provide high performance algorithms for:

- Recommenders
- Data matching
- Deep learning
- Sentiment analysis
- Churn prediction
- Personalization
- Object recognition
- Topic modeling
- Classification
- Clustering
- Regression
- Graph analytics
- Neural networks
- Matrix factorization
- Image processing
- Text analytics

- dato.com
- dato.com/products/create/open_source.html


madlib.net
MADlib
Home
Product
Documentation
Community
Download
MADlib: Big Data Machine Learning in SQL for Data Scientists
Open Source, commercially usable BSD license
Supports Postgres, Pivotal Greenplum Database, and Pivotal HAWQ
Powerful analytics for Big Data
Read More
madlib.net
Linear Regression
Linear regression can be used to model a linear relationship of a scalar dependent variable to one or more explanatory independent variables.
Latent Dirichlet Allocation
Latent Dirichlet Allocation is a topic modeling function used to identify recurring themes in a large document corpus.
Summary
The summary function provides summary statistics for any data table. These statistics include statistics such as: number of distinct values, number of missing values, mean, variance, min, max, most frequent values, quantiles, etc.
Logistic Regression
Logistic regression can be used to predict a binary outcome of a dependent variable from one or more explanatory independent variables.
Elastic Net Regularization
NEW
Elastic Net regularization is a regularization technique that can be implemented for either linear or logistic regression to help build a more robust model in the event of large numbers of explanatory independent variables.
Principal Component Analysis
Pricipal Component Analysis is a dimensional reduction technique that can be used to transform a high dimensional space into a lower dimensional space.
Apriori
Apriori, is a technique for evaluating frequent item-sets, which allows analysis of what events tend to occur together. For instance what items customers frequently purchase in a single transaction.
k-Means
k-Means is a clustering method used to identify regions of similarity within a dataset. It can be used for many types of analysis including customer segmentation analysis.

Flink | Overview | Quickstart | Features | Downloads | Documentation | FAQ

Apache Flink is an open source platform for distributed stream and batch data processing.

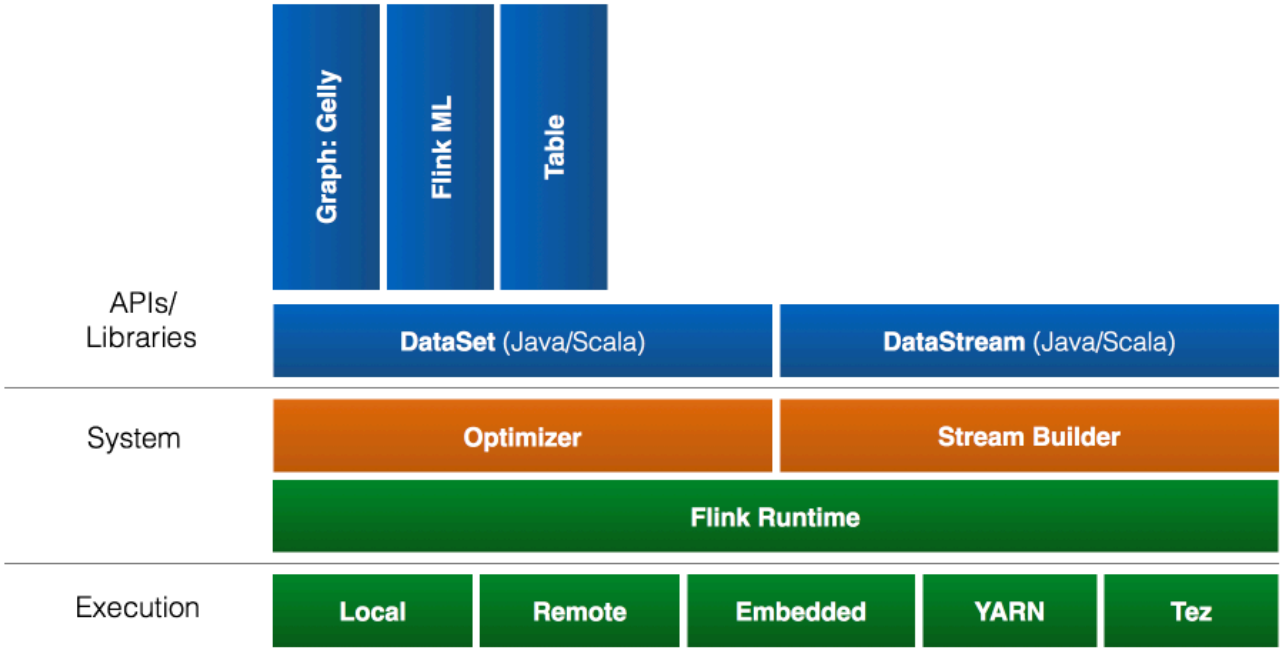
FlinkML

http://flink.apache.org

- Pipelines of transformers and learners
- Data pre-processing
  - **Feature scaling**
  - **Polynomial feature base mapper**
  - Feature hashing
  - Feature extraction for text
  - Dimensionality reduction
- Model selection and performance evaluation
  - Model evaluation using a variety of scoring functions
  - Cross-validation for model selection and evaluation
  - Hyper-parameter optimization
- Supervised learning
  - Optimization framework
    - **Stochastic Gradient Descent**
    - L-BFGS
  - Generalized Linear Models
    - **Multiple linear regression**
    - LASSO, Ridge regression
    - Multi-class Logistic regression
    - Random forests
    - **Support Vector Machines**
  - Decision trees
- Unsupervised learning
  - Clustering
    - K-means clustering
  - Principal Components Analysis
- Recommendation
  - **ALS**
- Text analytics
  - LDA
- Statistical estimation tools
- Distributed linear algebra
- Streaming ML

Stack

This is an overview of Flink's stack. Click on any component to go to the respective documentation.



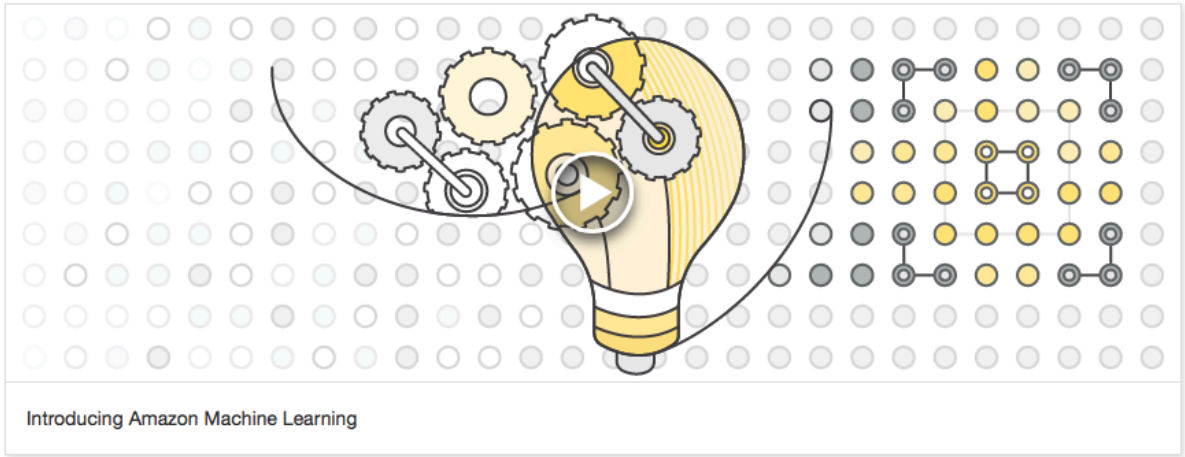
Amazon Machine Learning

Amazon Machine Learning is a service that makes it easy for developers of all skill levels to use machine learning technology. Amazon Machine Learning provides visualization tools and wizards that guide you through the process of creating machine learning (ML) models without having to learn complex ML algorithms and technology. Once your models are ready, Amazon Machine Learning makes it easy to get predictions for your application using simple APIs, without having to implement custom prediction generation code, or manage any infrastructure.

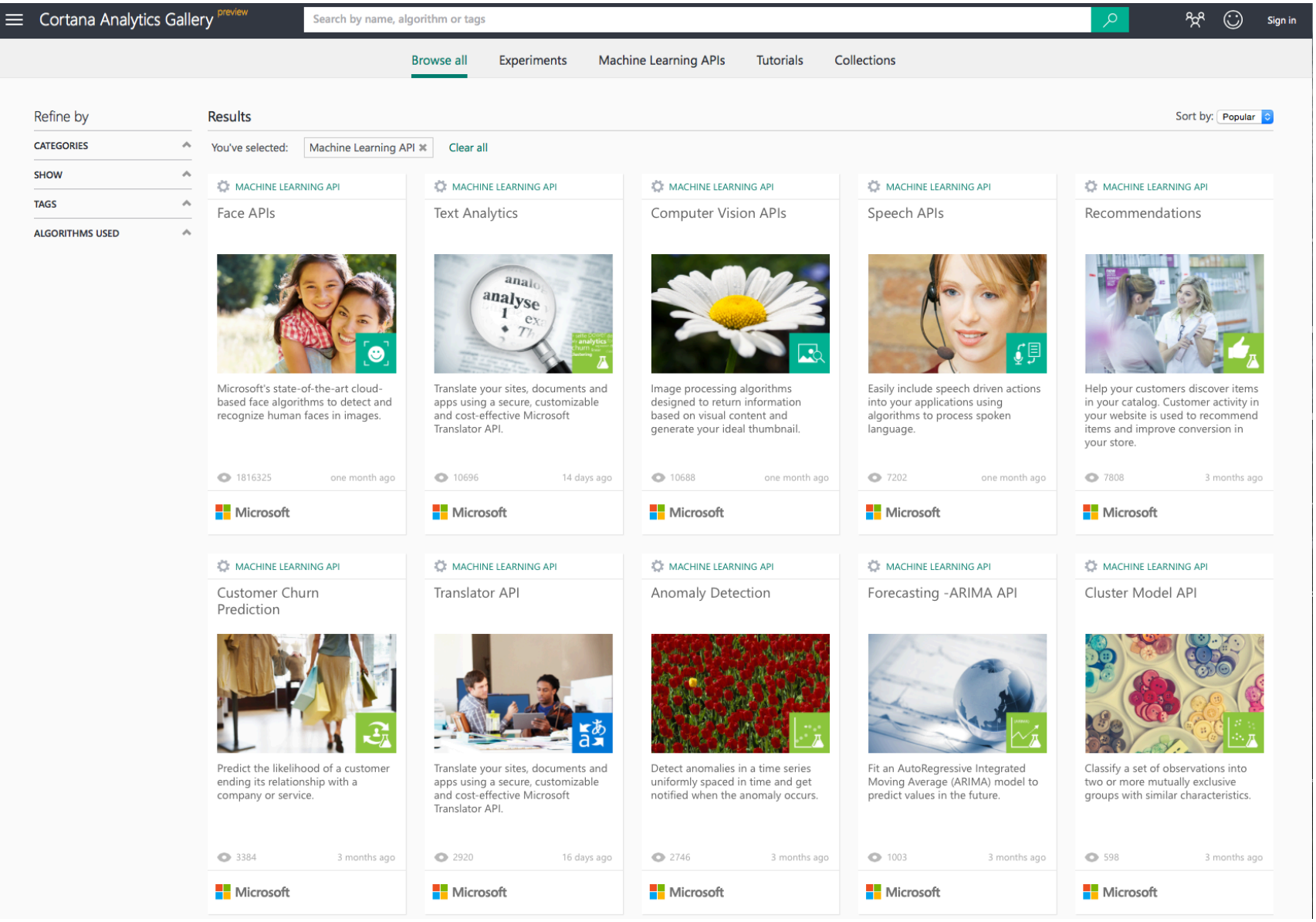
Amazon Machine Learning is based on the same proven, highly scalable, ML technology used for years by Amazon's internal data scientist community. The service uses powerful algorithms to create ML models by finding patterns in your existing data. Then, Amazon Machine Learning uses these models to process new data and generate predictions for your application.

Amazon Machine Learning is highly scalable and can generate billions of predictions daily, and serve those predictions in real-time and at high throughput. With Amazon Machine Learning, there is no upfront hardware or software investment, and you pay as you go, so you can start small and scale as your application grows.

aws.amazon.com/machine-learning/



Introducing Amazon Machine Learning



http://gallery.cortanaanalytics.com

Technische Universität Berlin

Popular Technologies Today

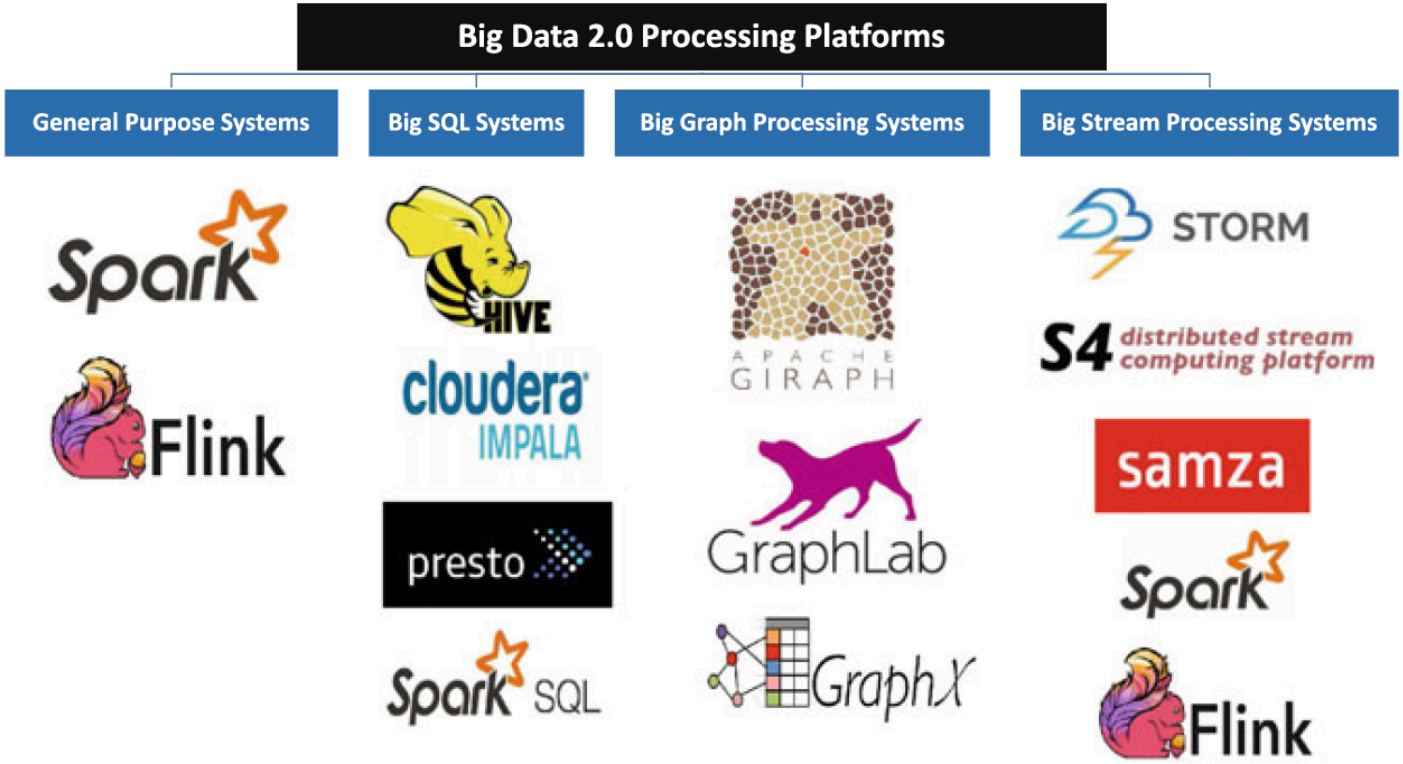


Fig. 1.7 Classification of Big Data 2.0 processing systems

Source: **Big Data 2.0 Processing Systems: A Survey** (Page 12), Springer, 2016

Technische Universität Berlin

hadoop.apache.org

Apache > Hadoop >

http://hadoop.apache.org

Top | Wiki

Search with Apache Solr | Search

Last Published: 02/13/2016 07:31:55

- About
  - Welcome
    - What Is Apache Hado...
    - Getting Started ...
    - Download Hadoop
    - Who Uses Hadoop?...
    - News
  - Releases
  - Mailing Lists
  - Issue Tracking
  - Who We Are?
  - Who Uses Hadoop?
  - Buy Stuff
  - Sponsorship
  - Thanks
  - Privacy Policy
  - Bylaws
  - License
- Documentation
- Related Projects

built with Apache Forrest

Welcome to Apache™ Hadoop®!

PDF

What Is Apache Hadoop?

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The project includes these modules:

- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

Other Hadoop-related projects at Apache include:

- **Ambari™**: A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters which includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig and Sqoop. Ambari also provides a dashboard for viewing cluster health such as heatmaps and ability to view MapReduce, Pig and Hive applications visually alongwith features to diagnose their performance characteristics in a user-friendly manner.
- **Avro™**: A data serialization system.
- **Cassandra™**: A scalable multi-master database with no single points of failure.
- **Chukwa™**: A data collection system for managing large distributed systems.
- **HBase™**: A scalable, distributed database that supports structured data storage for large tables.
- **Hive™**: A data warehouse infrastructure that provides data summarization and ad hoc querying.
- **Mahout™**: A Scalable machine learning and data mining library.
- **Pig™**: A high-level data-flow language and execution framework for parallel computation.
- **Spark™**: A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation.
- **Tez™**: A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases. Tez is being adopted by Hive™, Pig™ and other frameworks in the Hadoop ecosystem, and also by other commercial software (e.g. ETL tools), to replace Hadoop™ MapReduce as the underlying execution engine.
- **ZooKeeper™**: A high-performance coordination service for distributed applications.

Getting Started

To get started, begin here:

1. Learn about Hadoop by reading the documentation.
2. Download Hadoop from the release page.
3. Discuss Hadoop on the mailing list.

Download Hadoop

Technische Universität Berlin

http://flink.apache.org

flink.apache.org

Flink | Overview | Features | Downloads | FAQ | Quickstart | Documentation | Blog | Community | Project

Apache Flink is an open source platform for distributed stream and batch data processing.

Flink's core is a streaming dataflow engine that provides data distribution, communication, and fault tolerance for distributed computations over data streams.

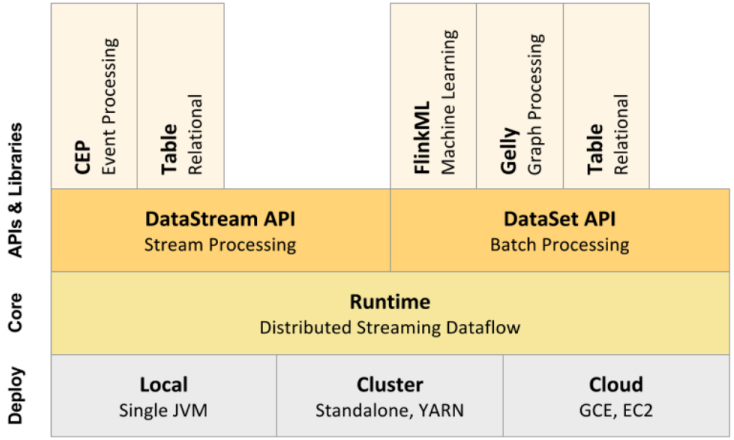
Flink includes **several APIs** for creating applications that use the Flink engine:

1. DataStream API for unbounded streams embedded in Java and Scala, and
2. DataSet API for static data embedded in Java, Scala, and Python,
3. Table API with a SQL-like expression language embedded in Java and Scala.

Flink also bundles **libraries for domain-specific use cases**:

1. CEP, a complex event processing library,
2. Machine Learning library, and
3. Gelly, a graph processing API and library.

You can **integrate** Flink easily with other well-known open source systems both for data input and output as well as deployment.



Streaming First

High throughput and low latency stream processing with exactly-once guarantees.

Batch on Streaming

Batch processing applications run efficiently as special cases of stream processing applications.

APIs, Libraries, and Ecosystem

DataSet, DataStream, and more. Integrated with the Apache Big Data stack.

Check out the Features page to get a tour of all major Flink features.

Getting Started

Latest blog posts

Technische Universität Berlin

http://spark.apache.org

spark.apache.org

Spark *Lightning-fast cluster computing*

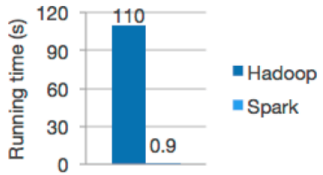
Download | Libraries | Documentation | Examples | Community | FAQ | Apache Software Foundation

Apache Spark™ is a fast and general engine for large-scale data processing.

Speed

Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

Spark has an advanced DAG execution engine that supports cyclic data flow and in-memory computing.



Logistic regression in Hadoop and Spark

Ease of Use

Write applications quickly in Java, Scala, Python, R.

Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it *interactively* from the Scala, Python and R shells.

```
text_file = spark.textFile("hdfs://...")

text_file.flatMap(lambda line: line.split())
    .map(lambda word: (word, 1))
    .reduceByKey(lambda a, b: a+b)
```

Word count in Spark's Python API

Latest News

Spark 1.6.1 released (Mar 09, 2016)

Submission is open for Spark Summit San Francisco (Feb 11, 2016)

Spark Summit East (Feb 16, 2016, New York) agenda posted (Jan 14, 2016)

Spark 1.6.0 released (Jan 04, 2016)

Archive

Download Spark

Built-in Libraries:

SQL and DataFrames
Spark Streaming
MLlib (machine learning)
GraphX (graph)

Third-Party Packages

Generality

Technische Universität Berlin

https://dato.com/products/create/

dato.com

Dato Create Intelligence™

PRODUCTS SOLUTIONS CUSTOMERS LEARN EVENTS BLOG

Buy Contact Us

FREE TRIAL

Home / Products / GraphLab Create

GraphLab Create

Sophisticated machine learning as easy as "Hello, World!"

TRY GRAPHLAB CREATE REQUEST A DEMO

Overview Features and Tech Specs Related Resources

GraphLab Create is an extensible machine learning framework that enables developers and data scientists to easily build and deploy intelligent applications and services at scale. It includes distributed data structures and rich libraries for data transformation and manipulation, scalable task-oriented machine learning toolkits for creating, evaluating, and improving machine learning models, data and model visualization for all aspects of development, and a client to define and deploy both distributed batch jobs to Dato Distributed™ as well as real-time machine learning services to Dato Predictive Services™. It is designed for end-to-end developer productivity, scale, and the variety and complexity of real-world data.

Create sophisticated models

Start with state-of-the-art toolkits, including implementations for deep learning, ranking-optimized factorization machines, topic modeling, graph analytics, linear models, clustering, and nearest neighbors. Then go as deep as you need, tuning exposed hyperparameters to customize your models.

Task-based Intelligence

Begin with your task rather than a research paper. Use human-intuitive machine learning abstractions in our toolkits. These toolkits are named for their use and offer default parameters and baseline models so your first application comes together fast.

Scale everything

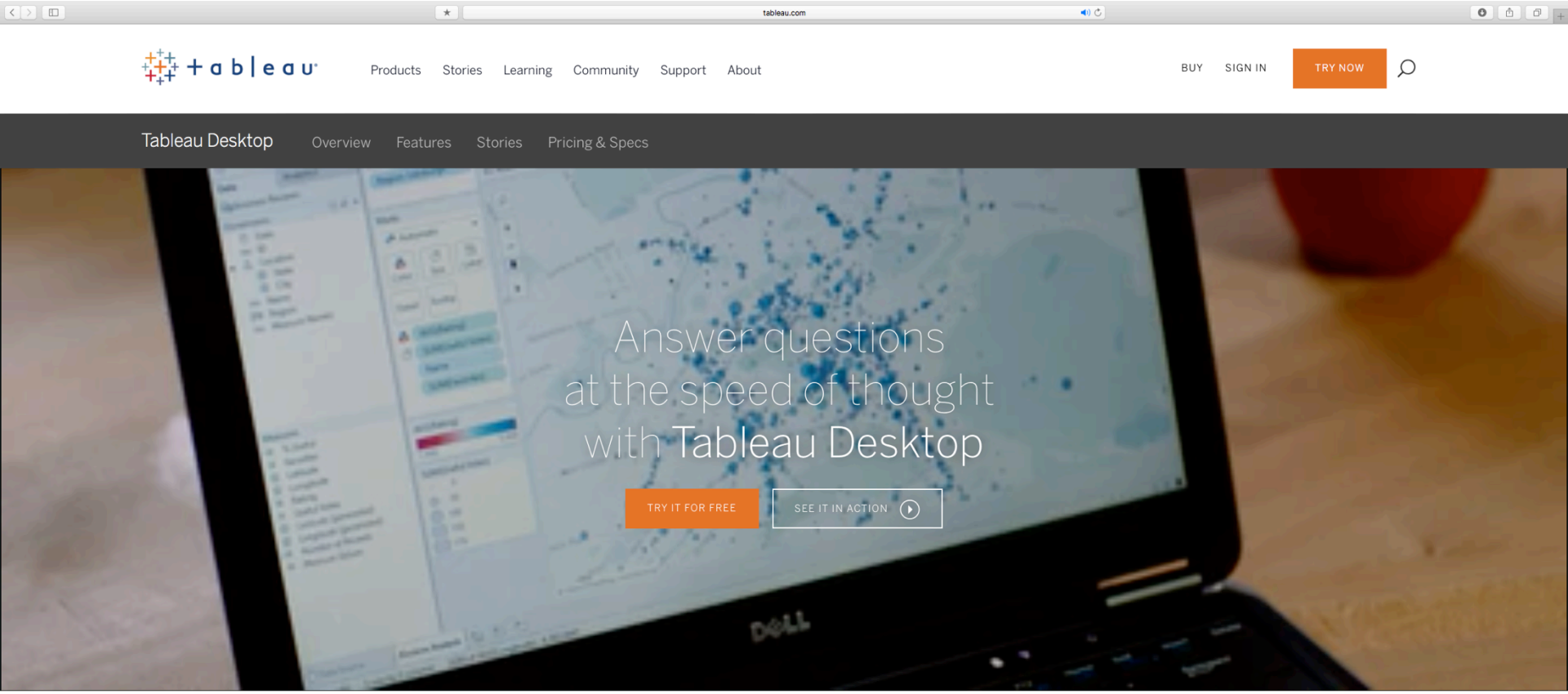
Interact with terabytes of data, blazingly fast, even on a laptop. SFrame, the out of core tabular data structure is columnar, distributed, and on-disk making is the most scalable data frame built for machine learning.

Explore and explain everything

GraphLab Create's visualization "Canvas" provides exploration and visualization of big data and sophisticated machine learning models. Easily explain your results.

Technische Universität Berlin

http://www.tableau.com/products/desktop



Analytics anyone can use

Technische Universität Berlin

7. Conclusion

Wrap-up

Technische Universität Berlin

Conclusion

- We have discussed data mining
  - definitions/processes, tasks, relevant issues, challenges
- In subsequent weeks, we will drill down and increasingly gain experience and familiarity with commonly used data analytics and their use in varying big data analytics platforms

Technische Universität Berlin

Reference

1. "*Successful Data Mining in Practice*," Richard De Veaux, Joint Statistical Meetings (JSM) 2003, San Francisco, California.
2. Data Mining Course, Ryan Tibshirani, Carnegie Mellon University
3. Scalable Data Analysis, David Blei, Princeton University
4. OpenML, Portal Listing ML Experiments, TU Eindhoven
5. Data Mining: Introductory & Advanced Topics, M. Dunham, Prentice Hall, 2003
6. Big Data: A Statistician's Perspective, statoo.com/BigDataDataScience/#oesg
7. Big Data and Its Technical Challenges, CACM July 2014 Vol. 57 No. 7
8. Data Mining on the Web, A. Rajaraman, J. Ullmann, Stanford University

Technische Universität Berlin

Homework

Technische Universität Berlin

Assignment

- Read **Chapter 1**, MMDS book
- Read the two reference articles described in the following slides
- Download Tableau
- Read Tableau documentation to gain familiarity

Technische Universität Berlin

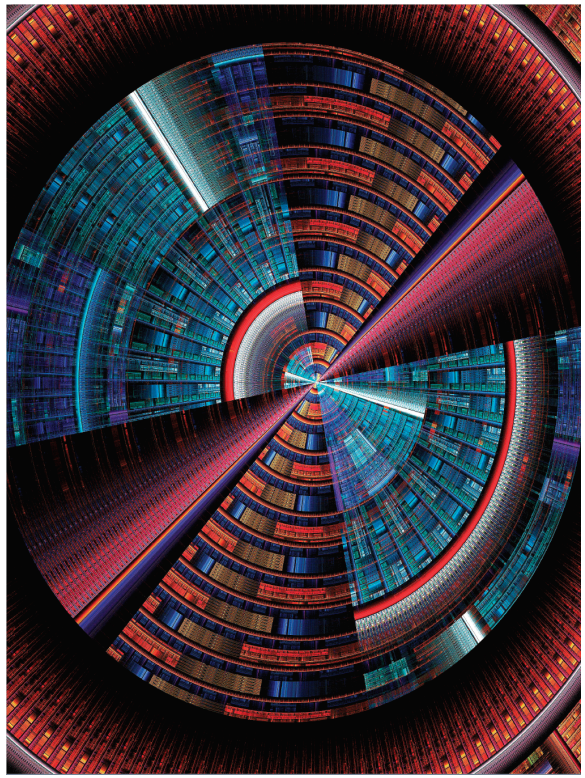
Assignment: Read This Article

DOI:10.1145/2611567

Exploring the inherent technical challenges in realizing the potential of Big Data.

BY H.V. JAGADISH, JOHANNES GEHRKE, ALEXANDROS LABRINIDIS, YANNIS PAPAKONSTANTINOU, JIGNESH M. PATEL, RAGHU RAMAKRISHNAN, AND CYRUS SHAHABI

Big Data and Its Technical Challenges



COMMUNICATIONS OF THE ACM | JULY 2014 | VOL. 57 | NO. 7

Assignment: Read These Articles As Well

Technische Universität Berlin

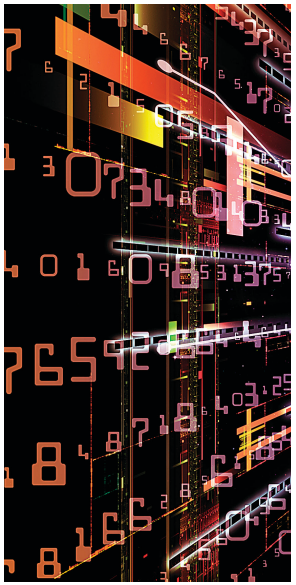
review articles

DOI:10.1145/2347736.2347755

Tapping into the "folk knowledge" needed to advance machine learning applications.

BY PEDRO DOMINGOS

A Few Useful Things to Know About Machine Learning



MACHINE LEARNING SYSTEMS automatically learn programs from data. This is often a very attractive alternative to manually constructing them, and in the last decade the use of machine learning has spread rapidly throughout computer science and beyond. Machine learning is used in Web search, spam filters, recommender systems, ad placement, credit scoring, fraud detection, stock trading, drug design, and many other applications. A recent report from the McKinsey Global Institute asserts that machine learning (a.k.a. data mining or predictive analytics) will be the driver of the next big wave of innovation.[15] Several fine textbooks are available to interested practitioners and researchers (for example, Mitchell[16] and Witten et al.[24]). However, much of the "folk knowledge" that is needed to successfully develop machine learning applications is not readily available in them. As a result, many machine learning projects take much longer than necessary or wind up producing less-than-ideal results. Yet much of this folk knowledge is fairly easy to communicate. This is the purpose of this article.

» key insights

- **Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. This is often feasible and cost-effective where manual programming is not. As more data becomes available, more ambitious problems can be tackled.**
- **Machine learning is widely used in computer science and other fields. However, developing successful machine learning applications requires a substantial amount of "black art" that is difficult to find in textbooks.**
- **This article summarizes 12 key lessons that machine learning researchers and practitioners have learned. These include pitfalls to avoid, important issues to focus on, and answers to common questions.**

Technische Universität Berlin

tableau Activation

- Below is a website (landing page link) for your upcoming class. Each student should go to the landing page, download Tableau, and enter the key noted below. This key will activate enough licenses for your entire class for the duration of the course.
- Download the latest version of Tableau, Desktop Key: **TDQ7-53EF-C9C0-B431-1AED**
- Click on the link above and select Get Started. On the form, enter your university email address for “Business Email”; and under "Organization", enter the name of your school, **Technische Universität Berlin.**
- Upon key expiration, students can request a one-year Desktop key via the **Tableau for Students** program: www.tableau.com/students.
- Each year a full-time student is enrolled, they can request a new one-year key.

Technische Universität Berlin

Tableau Contest

- Student Viz Assignment Contest, submit a viz created for an assignment, they'll be entered to win a Tableau swag bag. To enter:

  1. Make sure you're eligible to participate. You must be a student at an accredited academic institution. Check out the official contest rules here.

  2. Use a data set of your choice to create a viz. Need help finding data? Check out our student resource page for suggestions.

  3. Publish to your Tableau Public Profile, then submit the URL (make sure the data can be shared publicly).

- Entries will be judged by our very own Tableau Public team based on creativity, analytical depth, and beauty/design to determine the top 3 winners. The deadline to submit an entry is **May 16, 2017 at 11:59 p.m. PT**. One entry per student is allowed.