Scalable generalized linear models

Mauricio A. Álvarez, PhD

Scalable Machine Learning,
University of Sheffield

The
University
Of
Sheffield.

Contents

The exponential family

Generalized linear models

Iteratively reweighted least squares (IRLS)

Generalized linear models in Spark ML

References

Contents

The exponential family

Generalized linear models

Iteratively reweighted least squares (IRLS)

Generalized linear models in Spark ML

References

Introduction

- You are probably familiar by now with different types of probability distributions: the Gaussian, the Bernoulli, the Poisson, the Gamma, etc.

- It turns out that most of these are members of a broader class of distributions known as the exponential family.

Why the exponential family is important?

- It can be shown that the exponential family is the only family of distributions with finite-sized sufficient statistics.

- The exponential family is the only family of distributions for which conjugate priors exist.

- The exponential family can be shown to be the family of distributions that makes the least set of assumptions subject to some user-chosen constraints.

- The exponential family is at the core of generalized linear models.

Definition

- It is said that a pdf or a pmf $p(\mathbf{x}|\boldsymbol{\theta})$, with $\mathbf{x} \in \mathbb{R}^p$ and $\boldsymbol{\theta} \in \mathbb{R}^d$, is in the **exponential family** if it is of the form

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} h(\mathbf{x}) \exp \left[\boldsymbol{\theta}^\top \phi(\mathbf{x})\right],$$

  where

$$Z(\boldsymbol{\theta}) = \int h(\mathbf{x}) \exp \left[\boldsymbol{\theta}^\top \phi(\mathbf{x})\right] d\mathbf{x}.$$

- $\boldsymbol{\theta}$ are known as the **natural parameters** or **canonical parameters**.

- $\phi(\mathbf{x}) \in \mathbb{R}^d$ is called a vector of **sufficient statistics**.

- $Z(\boldsymbol{\theta})$ is known as the **partition function**.

- $h(\mathbf{x})$ is a scaling constant, often 1.

Definition

- Distributions in the exponential family can also be expressed as

$$p(\mathbf{x}|\boldsymbol{\theta}) = h(\mathbf{x}) \exp\left[\boldsymbol{\theta}^\top \phi(\mathbf{x}) - A(\boldsymbol{\theta})\right],$$

  where

$$A(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta}).$$

- $A(\boldsymbol{\theta})$ is called the **log partition function** or **cumulant function**.

- If $\phi(\mathbf{x}) = \mathbf{x}$, we say it is a **natural exponential family**.

Example: Bernoulli (I)

- For the Bernoulli distribution, $x \in \{0, 1\}$, and we have

$$p(x|\mu) = \mathrm{Bern}(x|\mu) = \mu^x (1-\mu)^{1-x},$$

where $\mu = p(x = 1)$.

- The expected value of x is equal to $\mathbb{E}[x] = \mu$.

- The distribution above can be written as

$$\begin{aligned} p(x|\mu) &= \exp\{x \log \mu + (1-x)\log(1-\mu)\}, \\ &= (1-\mu)\exp\left\{\log\left(\frac{\mu}{1-\mu}\right) x\right\} \end{aligned}$$

Example: Bernoulli (II)

- Comparing terms with the general expression for the exponential family, we observe that

$$\theta = \log\left(\frac{\mu}{1-\mu}\right), \quad \mu = \sigma(\theta) = \frac{1}{1+\exp(-\theta)}.$$

- This means

$$Z(\theta) = \frac{1}{1-\mu} = \frac{1}{1-\frac{1}{1+\exp(-\theta)}} = \frac{1}{\frac{\exp(-\theta)}{1+\exp(-\theta)}} = \frac{1}{\frac{1}{\exp(\theta)+1}} = \frac{1}{\sigma(-\theta)}.$$

- The Bernoulli distribution can be written as $p(x|\theta) = \sigma(-\theta)\exp(\theta x)$.

- The term $\theta = \log\left(\frac{\mu}{1-\mu}\right)$ is known as the **log-odds ratio**.

Example: univariate Gaussiana distribution

- The univariate Gaussian can be written in exponential family form as

$$
\begin{aligned}
\mathcal{N}(x|\mu,\sigma^2) &= \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left[-\frac{1}{2\sigma^2}(x-\mu)^2\right] \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left[-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}\mu^2\right] \\
&= \frac{1}{Z(\boldsymbol{\theta})} \exp\left(\boldsymbol{\theta}^\top \phi(x)\right),
\end{aligned}
$$

where

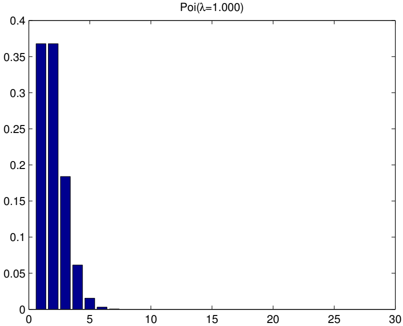
$$
\begin{aligned}
\boldsymbol{\theta} &= \begin{bmatrix} \mu/\sigma^2 \\ -\frac{1}{2\sigma^2} \end{bmatrix}, \qquad \phi(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}, \\
Z(\boldsymbol{\theta}) &= \sqrt{2\pi}\sigma \exp\left\{\frac{\mu^2}{2\sigma^2}\right\}, \\
A(\boldsymbol{\theta}) &= -\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2) - \frac{1}{2}\log(2\pi).
\end{aligned}
$$

Example: Poisson distribution (I)

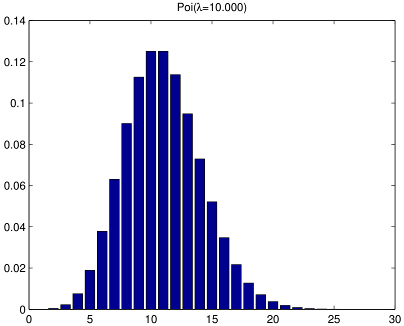
- The Poisson distribution follows as

$$\mathrm{Poi}(x|\lambda) = \exp(-\lambda)\frac{\lambda^x}{x!},$$

where $\lambda > 0$, and $x \in \{0, 1, 2, \ldots\}$.



$\lambda = 1$



$\lambda = 10$

Example: Poisson distribution (II)

- As a member of the exponential family, it can be written as

$$\mathrm{Poi}(x|\lambda) = \frac{h(x)}{Z(\theta)} \exp(\theta x),$$

where $\theta = \log \lambda$, $h(x) = 1/x!$, and $Z(\theta) = \exp(\lambda)$. Also, $A(\theta) = \lambda$.

- Recall that the expected value of x is equal to $\mathbb{E}[x] = \lambda$.

- Then, the mean parameter λ can be recovered from the canonical parameter using

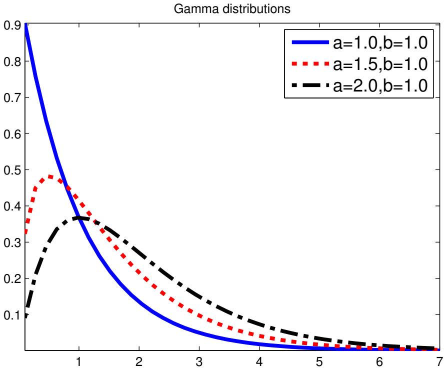
$$\lambda = \exp(\theta).$$

Example: Gamma distribution (I)

- The Gamma distribution follows as

$$\mathrm{Ga}(x|a,b) = \frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx),$$

where $a > 0$ (shape), and $b > 0$ (rate). $\Gamma(a) = \int_0^\infty u^{a-1} e^{-u} du$ is the Gamma function.



Example: Gamma distribution (II)

- As a member of the exponential family, it can be written as

$$\mathrm{Ga}(x|a,b) = \frac{1}{Z(\boldsymbol{\theta})} \exp(\boldsymbol{\theta}^\top \phi(x)),$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} a-1 \\ -b \end{bmatrix}, \qquad \phi(x) = \begin{bmatrix} \log x \\ x \end{bmatrix},$$
$$Z(\boldsymbol{\theta}) = \frac{\Gamma(a)}{b^a}, \quad A(\boldsymbol{\theta}) = \log\Gamma(a) - a\log b.$$

Contents

The exponential family

Generalized linear models

Iteratively reweighted least squares (IRLS)

Generalized linear models in Spark ML

References

Definition

- Linear and logistic regression are examples of generalized linear models, or GLM.

- These are models in which the output density is in the exponential family.

- The mean parameters are a linear combination of the inputs, passed through a possibly nonlinear function, such as the logistic function.

General form (I)

- We previously used x as the variable we were modelling.
- Here we use y since the variable we want to model is the response variable.
- We want to model the relationship between a response variable y_i, and an input vector $\mathbf{x}_i$.
- Let us first consider the case of an unconditional distribution for the response variable

$$p(y_i|\theta, \sigma^2) = \exp\left[\frac{y_i\theta - A(\theta)}{\sigma^2} + c(y_i, \sigma^2)\right],$$

  where σ^2 is the **dispersion parameter**, θ is the natural parameter, A is the partition function, and c is the normalisation constant. Usually, $\sigma^2 = 1$.
- The expression for $p(y_i|\theta, \sigma^2)$ looks similar to the exponential family.

General form (II)

- For example, in logistic regression, θ is the log-odds ratio

$$\theta = \log\left(\frac{\mu}{1-\mu}\right),$$

where $\mu = \mathbb{E}[y] = P(y = 1)$ is the mean parameter.

- To convert from the mean parameter to the natural parameter, we can use a function ψ, $\theta = \psi(\mu)$.

- ψ is uniquely determined by the form of the exponential family distribution.

- The mapping is invertible, so that $\mu = \psi^{-1}(\theta)$.

General form (III)

- Now let us add inputs or covariates.

- We first define a linear function of the inputs $\eta_i = \mathbf{w}^\top \mathbf{x}_i$.

- We now make the mean of the distribution be some invertible monotonic function of this linear combination.

- By convention, this function, known as the **mean function**, is denoted by g^{-1}, so

$$\mu_i = g^{-1}(\eta_i) = g^{-1}(\mathbf{w}^\top \mathbf{x}_i).$$

- The inverse of the mean function, namely $g()$, is called the **link function**.

Relationships between functions

$$\begin{tikzcd}
\boldsymbol{w} \arrow[dr] & & & \\
& \eta_i \arrow[r, shift left, "g^{-1}"] & \mu_i \arrow[l, shift left, "g"] \arrow[r, shift left, "\Psi"] & \theta_i \arrow[l, shift left, "\Psi^{-1}"] \\
\boldsymbol{x}_i \arrow[ur] & & &
\end{tikzcd}$$

$g^{-1}()$ is the **mean function.** $g()$ is the **link function.**

Link function

- We are free to choose almost any function we like for g, so long as it is invertible, and so long as g^{-1} has the appropriate range.

- For example, in logistic regression, we set $\mu_i = g^{-1}(\eta_i) = \sigma(\eta_i)$.

GLM with canonical link function

- One particularly simple form of link function is to use $g = \psi$.

- This is called the **canonical link function.**

- In this case $\theta_i = \eta_i = \mathbf{w}^\top \mathbf{x}_i$, so the model becomes

$$p(y_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) = \exp\left[\frac{y_i \mathbf{w}^\top \mathbf{x}_i - A(\mathbf{w}^\top \mathbf{x}_i)}{\sigma^2} + c(y_i, \sigma^2)\right].$$

Canonical link functions $g = \psi$ for some GLMs

Distribution	Link $g(\mu)$	$\theta = \psi(\mu)$	$\mu = \psi^{-1}(\theta)$
$\mathcal{N}(\mu, \sigma^2)$	identity	$\theta = \mu$	$\mu = \theta$
$\mathrm{Ber}(\mu)$	logit	$\theta = \log\left(\frac{\mu}{1-\mu}\right)$	$\mu = \sigma(\theta)$
$\mathrm{Poi}(\mu)$	log	$\theta = \log(\mu)$	$\mu = \exp(\theta)$
$\mathrm{Ga}(a, b)$	inverse	$\theta = \mu^{-1}$	$\mu = \theta^{-1}$.

Mean and variance of the response variable

- It can be shown that

$$\begin{aligned} \mathbb{E}[y|\mathbf{x}_i, \mathbf{w}, \sigma^2] &= \mu_i = A'(\theta_i) \\ \text{var}[y|\mathbf{x}_i, \mathbf{w}, \sigma^2] &= \sigma_i^2 = A''(\theta_i)\sigma^2. \end{aligned}$$

Example: linear regression

- In linear regression, the response variable follows a normal distribution,

$$\begin{aligned} p(y_i|\mu_i,\sigma^2) &= \mathcal{N}(y_i|\mu_i,\sigma^2) \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}}\exp\left[-\frac{1}{2\sigma^2}(y_i-\mu_i)^2\right] \\ &= \exp\left[\frac{y_i\mu_i-\frac{\mu_i^2}{2}}{\sigma^2}-\frac{y_i^2}{2\sigma^2}+\log\left(\frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}}\right)\right] \\ &= \exp\left[\frac{y_i\mu_i-\frac{\mu_i^2}{2}}{\sigma^2}-\frac{1}{2}\left(\frac{y_i^2}{\sigma^2}+\log(2\pi\sigma^2)\right)\right]. \end{aligned}$$

- For linear regression, $y_i \in \mathbb{R}$.

- The link function is the identity $\theta_i = \mu_i = \mathbf{w}^\top \mathbf{x}_i$.

- With $A(\mu_i) = \mu_i^2/2$, $\mathbb{E}[y_i] = \mu_i$, and $\mathrm{var}[y_i] = \sigma^2$.

Example: logistic regression

- In logistic regression, the response variable follows a Bernoulli distribution

$$\begin{aligned} p(y_i|\mu_i, \sigma^2) &= \mu_i^{y_i}(1-\mu_i)^{y_i} \\ &= \exp\left[\log\left(\frac{\mu_i}{1-\mu_i}\right) y_i - (-\log(1-\mu_i))\right]. \end{aligned}$$

- The link function is the logit, $\log\left(\frac{\mu_i}{1-\mu_i}\right) = \mathbf{w}^\top \mathbf{x}_i$.

- With $A(\theta_i) = -\log(1-\sigma(\theta_i))$, $\mathbb{E}[y_i] = \sigma(\theta_i)$, and $\text{var}[y_i] = \sigma(\theta_i)(1-\sigma(\theta_i))$.

Example: Poisson regression

- In Poisson regression, the response variable follows a Poisson distribution

$$p(y_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) = \exp[y_i \log(\mu_i) - \mu_i - \log(y_i!)].$$

- The link function is log, $\log \mu_i = \mathbf{w}^\top \mathbf{x}_i$.

- With $A(\theta_i) = \exp(\theta_i)$, $\mathbb{E}[y_i] = \exp(\theta_i) = \lambda_i$.

Contents

The exponential family

Generalized linear models

Iteratively reweighted least squares (IRLS)

Generalized linear models in Spark ML

References

Log-likelihood for a GLM

- One of the appealing properties of GLMs is that they can be fit using exactly the same methods that we used to fit logistic regression.

- In particular, the log-likelihood has the following form

$$\ell(\mathbf{w}) = \frac{1}{\sigma^2}\sum_{i=1}^{N}\ell_i = \frac{1}{\sigma^2}\sum_{i=1}^{N}[\theta_i y_i - A(\theta_i)],$$

where $\ell_i = \theta_i y_i - A(\theta_i)$.

Gradient for the log-likelihood

- We can compute the gradient vector using the chain rule as follows

$$\begin{aligned}\frac{\partial \ell_i}{\partial w_j} &= \frac{d\ell_i}{d\theta_i}\frac{d\theta_i}{d\mu_i}\frac{d\mu_i}{d\eta_i}\frac{\partial \eta_i}{\partial w_j} \\ &= (y_i - A'(\theta_i))\frac{d\theta_i}{d\mu_i}\frac{d\mu_i}{d\eta_i}x_{i,j} \\ &= (y_i - \mu_i)\frac{d\theta_i}{d\mu_i}\frac{d\mu_i}{d\eta_i}x_{i,j}.\end{aligned}$$

- If we use a canonical link, $\theta_i = \eta_i$, this simplifies to

$$\mathbf{g}(\mathbf{w}) = \frac{1}{\sigma^2}\left[\sum_{i=1}^{N}(y_i - \mu_i)\mathbf{x}_i\right] = \frac{1}{\sigma^2}\mathbf{X}^\top(\mathbf{y} - \boldsymbol{\mu}),$$

where $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_N]^\top$.

- This can be used inside a (stochastic) gradient descent procedure.

Hessian for the log-likelihood

- For improved efficiency, we could use a second-order method.

- If we use a canonical link, the Hessian is given by

$$\mathbf{H}(\mathbf{w}) = -\frac{1}{\sigma^2}\sum_{i=1}^{N}\frac{d\mu_i}{d\theta_i}\mathbf{x}_i\mathbf{x}_i^\top = -\frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{\Sigma}\mathbf{X},$$

  where $\mathbf{\Sigma} = \mathrm{diag}\left(\frac{d\mu_1}{d\theta_1}, \cdots, \frac{d\mu_N}{d\theta_N}\right)$.

- This can be used inside the Iterative Reweighted Least Squares (IRLS) algorithm.

Iterative reweighted least squares (IRLS) algorithm

- The Iterative Reweighted Least Squares algorithm is a particular case of the Newton's method.

- The updated parameters are obtained by iteratively solving a weighted least squares problem.

A least squares problem

- Remember that a least squares (*LS*) problem refers to

$$LS(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2,$$

  for a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N} = \{\mathbf{X}, \mathbf{y}\}$.

- It can be shown that the vector **w** that minimises $LS(\mathbf{w})$ is given as

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

A weighted least squares problem

- A weighted least squares (*WLS*) problem refers to

$$WLS(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^{N} r_i (y_i - \mathbf{w}^\top \mathbf{x}_i)^2,$$

  for a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i, r_i\}_{i=1}^{N} = \{\mathbf{X}, \mathbf{R}, \mathbf{y}\}$, with $\mathbf{R} = \mathrm{diag}(r_1, \ldots, r_N)$.

- It can be shown that the vector **w** that minimises $WLS(\mathbf{w})$ is given as

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{R} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{R} \mathbf{y}.$$

Iterative reweighted least squares problem

- Newton's method for the log-likelihood of the GLM follows as

$$\begin{aligned}\mathbf{w}_{k+1} &= \mathbf{w}_k - \mathbf{H}_k^{-1}\mathbf{g}_k \\ &= \mathbf{w}_k + (\mathbf{X}^\top \Sigma_k \mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{y} - \boldsymbol{\mu}_k) \\ &= (\mathbf{X}^\top \Sigma_k \mathbf{X})^{-1}[\mathbf{X}^\top \Sigma_k \mathbf{X}\mathbf{w}_k + \mathbf{X}^\top(\mathbf{y} - \boldsymbol{\mu}_k)] \\ &= (\mathbf{X}^\top \Sigma_k \mathbf{X})^{-1}\mathbf{X}^\top \Sigma_k \mathbf{z}_k,\end{aligned}$$

  where $\mathbf{z}_k = \mathbf{X}\mathbf{w}_k + \Sigma_k^{-1}(\mathbf{y} - \boldsymbol{\mu}_k)$ is known as the **working response**.

- At iteration k, the solution for $\mathbf{w}_{k+1}$ has a similar form to the solution for a weighted least squared problem replacing $\mathbf{R}$ for Σ_k, and $\mathbf{y}$ for $\mathbf{z}_k$.

- The name IRLS is due to at each iteration, we solve a weighted least squares problem, where the weight matrix Σ_k changes at each iteration.

Contents

The exponential family

Generalized linear models

Iteratively reweighted least squares (IRLS)

Generalized linear models in Spark ML

References

GeneralizedLinearRegression()

- It uses IRLS (Iterative Reweighted Least Squares) for optimisation.

- It only allows ℓ_2 regularisation.

- Spark currently only supports up to 4096 features through its `GeneralizedLinearRegression` interface.

- It will throw an exception if this constraint is exceeded.

GLM available in Spark

- It includes the following families

Family	Response type	Supported links
Gaussian	Continuous	Identity*, Log, Inverse
Binomial	Binary	Logit*, Probit, CLogLog
Poisson	Count	Log*, Identity, Sqrt
Gamma	Continuous	Inverse*, Identity, Log
Tweedie	Zero-inflated continuous	Power link function

  where * stands for canonical link.

- For any random variable Z that obeys a Tweedie distribution, the variance $\text{var}(Z)$ relates to the mean $\mathbb{E}(Z)$ by the power law: $\text{var}(Z) = a\mathbb{E}(Z)^p$, where a and p are positive constants.

- The parameters are set using `family` and `link`.

Parameters to adjust

- **`maxIter`**: max number of iterations.
- **`regParam`**: regularization parameter (≥ 0).
- **`family`**: name of family which is a description of the label distribution to be used in the model.
- **`link`**: name of link function which provides the relationship between the linear predictor and the mean of the distribution function.

Particular cases: `LinearRegression()`

- If your ‘family’ is Gaussian and the link function is the ‘identity’, your model is just equivalent to linear regression.

- My recommendation is that you use `LinearRegression()` instead of `GeneralizedLinearRegression()`.

- `LinearRegression()` allows for ℓ_1, ℓ_2 and elastic net regularization through L-BFGS or OWL-QN.

Particular cases: `LogisticRegression()`

- If your ‘family’ is Binomial and the link function is ‘logit’, your model is equivalent to logistic regression.

- My recommendation is that you use `LogisticRegression()` instead of `GeneralizedLinearRegression()`.

- `LogisticRegression()` allows for ℓ_1, ℓ_2 and elastic net regularization through L-BFGS or OWL-QN.

Contents

The exponential family

Generalized linear models

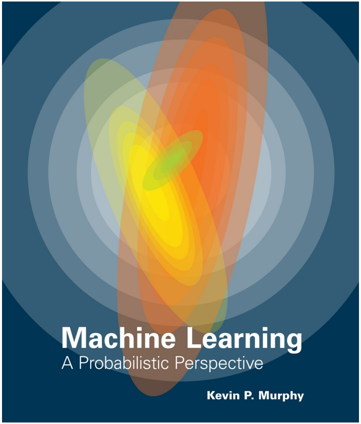
Iteratively reweighted least squares (IRLS)

Generalized linear models in Spark ML

References

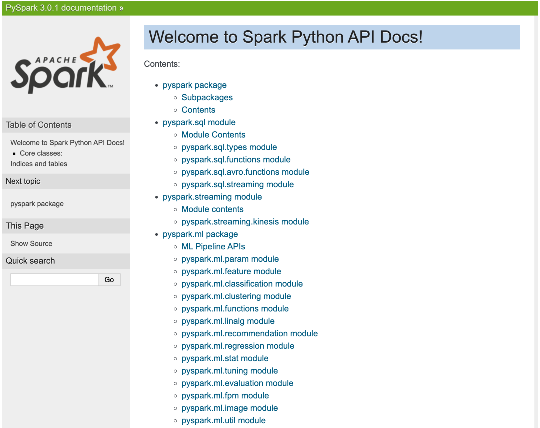
References used in this lecture

Book: *Machine Learning: A Probabilistic Perspective* by Kevin P Murphy, 2012.



References used in this lecture

Website: *Spark Python API Documentation*.



PySpark 3.0.1 documentation »

Welcome to Spark Python API Docs!

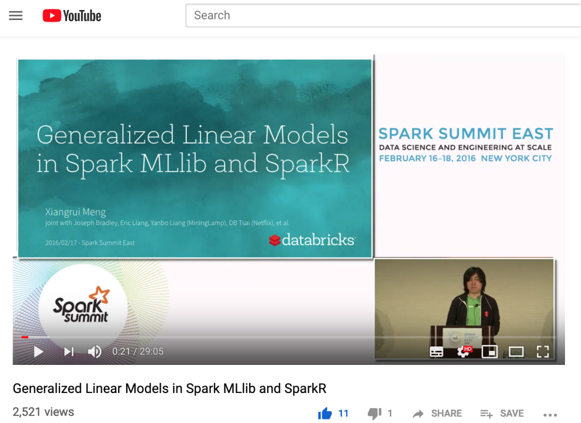
Contents:

- pyspark package
  - Subpackages
  - Contents
- pyspark.sql module
  - Module Contents
  - pyspark.sql.types module
  - pyspark.sql.functions module
  - pyspark.sql.avro.functions module
  - pyspark.sql.streaming module
- pyspark.streaming module
  - Module contents
  - pyspark.streaming.kinesis module
- pyspark.ml package
  - ML Pipeline APIs
  - pyspark.ml.param module
  - pyspark.ml.feature module
  - pyspark.ml.classification module
  - pyspark.ml.clustering module
  - pyspark.ml.functions module
  - pyspark.ml.linalg module
  - pyspark.ml.recommendation module
  - pyspark.ml.regression module
  - pyspark.ml.stat module
  - pyspark.ml.tuning module
  - pyspark.ml.evaluation module
  - pyspark.ml.fpm module
  - pyspark.ml.image module
  - pyspark.ml.util module

`https://spark.apache.org/docs/3.0.1/api/python/index`

References used in this lecture

Youtube video: *Generalized Linear Models in Spark MLlib and SparkR* by Xiangrui Meng (from Databricks).


YouTube
Search
Generalized Linear Models in Spark MLlib and SparkR
Xiangrui Meng
joint with Joseph Bradley, Eric Liang, Yanbo Liang (MiningLamp), DB Tsai (Netflix), et al.
2016/02/17 - Spark Summit East
databricks
SPARK SUMMIT EAST
DATA SCIENCE AND ENGINEERING AT SCALE
FEBRUARY 16-18, 2016 NEW YORK CITY
Spark summit
0:21 / 29:05
HD

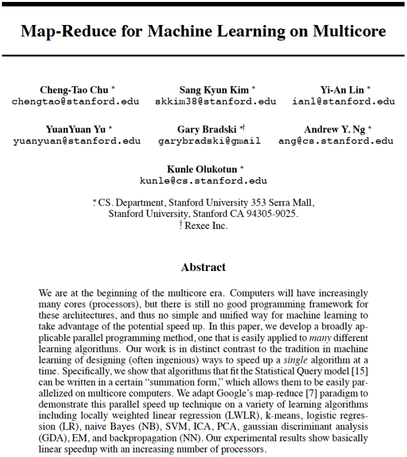
Generalized Linear Models in Spark MLlib and SparkR

2,521 views

`https://www.youtube.com/watch?v=PSZW6hcQ_7w`

References used in this lecture

Paper: *Map-Reduce for Machine Learning on Multicore* by C-T Chu et al. (2007).



Map-Reduce for Machine Learning on Multicore

Cheng-Tao Chu *
chengtao@stanford.edu

Sang Kyun Kim *
skkim38@stanford.edu

Yi-An Lin *
ianl@stanford.edu

YuanYuan Yu *
yuanyuan@stanford.edu

Gary Bradski *†
garybradski@gmail

Andrew Y. Ng *
ang@cs.stanford.edu

Kunle Olukotun *
kunle@cs.stanford.edu

* CS. Department, Stanford University 353 Serra Mall,
Stanford University, Stanford CA 94305-9025.
† Rexee Inc.

Abstract

We are at the beginning of the multicore era. Computers will have increasingly many cores (processors), but there is still no good programming framework for these architectures, and thus no simple and unified way for machine learning to take advantage of the potential speed up. In this paper, we develop a broadly applicable parallel programming method, one that is easily applied to *many* different learning algorithms. Our work is in distinct contrast to the tradition in machine learning of designing (often ingenious) ways to speed up a *single* algorithm at a time. Specifically, we show that algorithms that fit the Statistical Query model [15] can be written in a certain "summation form," which allows them to be easily parallelized on multicore computers. We adapt Google's map-reduce [7] paradigm to demonstrate this parallel speed up technique on a variety of learning algorithms including locally weighted linear regression (LWLR), k-means, logistic regression (LR), naive Bayes (NB), SVM, ICA, PCA, gaussian discriminant analysis (GDA), EM, and backpropagation (NN). Our experimental results show basically linear speedup with an increasing number of processors.