# Scalable Statistics with Large Datasets

Munich, 30.07.2020

**Authors:** Cheng-Wei Wang, Martin Rosenzweig, Nino Mumladze,
Tatiana Kuznetsova, Thomas Decker
**Mentors:** Aurélien Ouattara, Florian Felice, Liubomyr Bregman
Amazon
**Co-mentor:** M.Sc Özge Sahin (Department of Mathematics)
**Project Lead:** Dr. Ricardo Acevedo Cabra (Department of Mathematics)
**Supervisor:** Prof. Dr. Massimo Fornasier (Department of Mathematics)

# Presentation Plan

1. **Problem definition and goals of the project**

2. **Search for the best model specification:**

    1. **Pooled logistic and probit regressions in TensorFlow**
    2. **Unobserved effects in Apache Spark**

3. **Results**

# Problem Definition and Goals of the Project

**We can create:**
Statistical Inference ✓ Small data ✗
Statistical Inference ✗ Big data ✓

**Goal:**
Statistical Inference ✓
Big data ✓

## Complexity

o **open problem in econometrics**
o ensuring statistical consistency
o computational performance
o Robustness to singularity

**Develop an efficient scalable tool for statistical inference**

# Data Description

*Data Insights:*

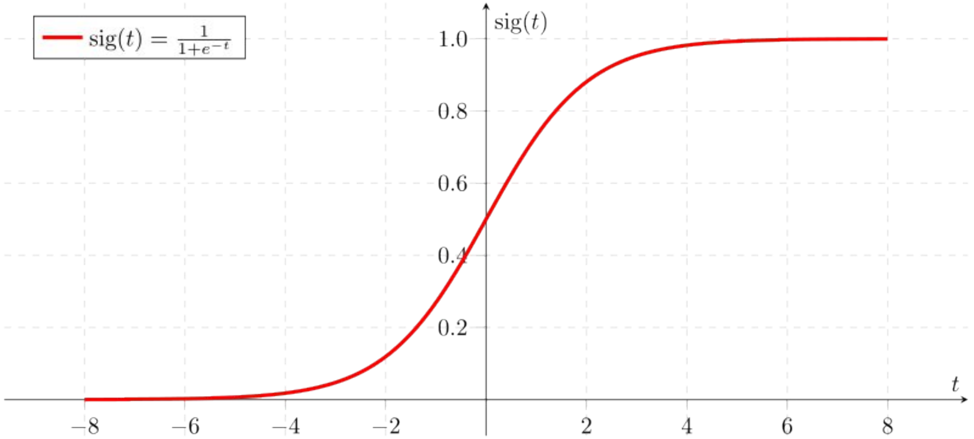
28 features - 11 continuous, 1 categorical, 16 binary

Binary Outcome 500 M observations

Panel data

*Different approaches for binary Panel data:*

1. Pooled Logistic Regression
2. Pooled Probit Regression
3. Conditional MLE for fixed effects
4. Unconditional MLE for fixed effects



# Pooled vs Fixed Effects Logit

$$\mathbb{P}(Y_{it} = 1 | x_{it}, \beta) = \sigma(x_{it}^T \beta + \beta_0)$$

$$\mathbb{P}(Y_{it} = 1 | x_{it}, \beta, \alpha_i) = \sigma(x_{it}^T \beta + \alpha_i)$$

## Expected output:

**Average Partial Effects**

$$APE_j = \beta_j \frac{\sum f(\boldsymbol{x}^T \boldsymbol{\beta})}{N}$$

$$APE_j = \beta_j \frac{\sum_{i=1}^{I} \sum_{t=1}^{T_i} f(\boldsymbol{x}^T \boldsymbol{\beta} + \boldsymbol{\alpha}_i)}{N\overline{T}}$$

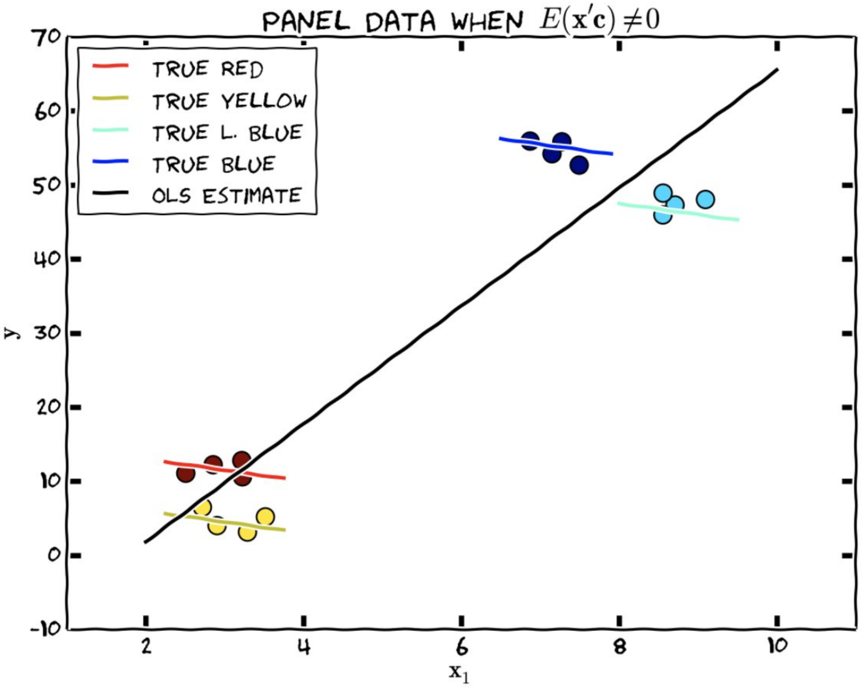
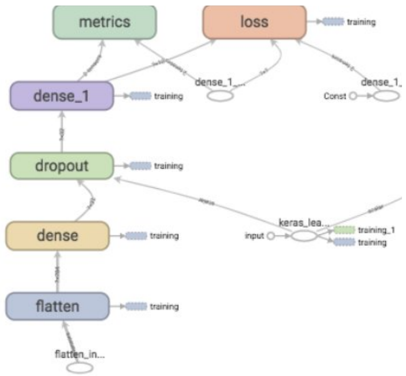



Figure 1. Unobserved Individual Heterogeneity and the Population Regression l

# We use the leading technologies in distributed computing




***GPUs*** *- TensorFlow:*

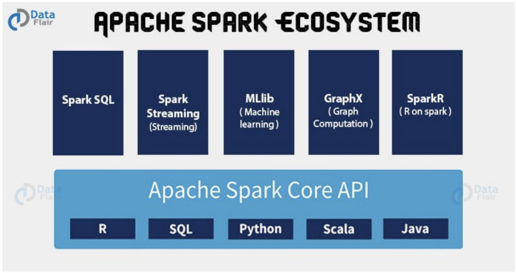
- ✓ open-source machine learning platform, Google
- ✓ initially developed for **large numerical computations**

- ✓ Seamless parallelization of computations across **several GPUs** in a batched manner
- ✓ Impossible to store the data in RAM ➔ efficient data pipelines

***Cluster-based computations*** *- Apache Spark:*

- ✓ open-source cluster-computing system
- ✓ one of the most popular **data preprocessing systems**

- ✓ fast, in-memory computing
- ✓ automatically distribute the data across the cluster and parallelize the operations we perform on them




Apache Spark Ecosystem — Spark Core, Spark SQL, Spark Streaming, MLlib, GraphX, SparkR.

# Pooled vs Fixed Effects Logit

$$\mathbb{P}(Y_{it} = 1 | x_{it}, \beta) = \sigma(x_{it}^T \beta + \beta_0)$$

$$\mathbb{P}(Y_{it} = 1 | x_{it}, \beta, \alpha_i) = \sigma(x_{it}^T \beta + \alpha_i)$$

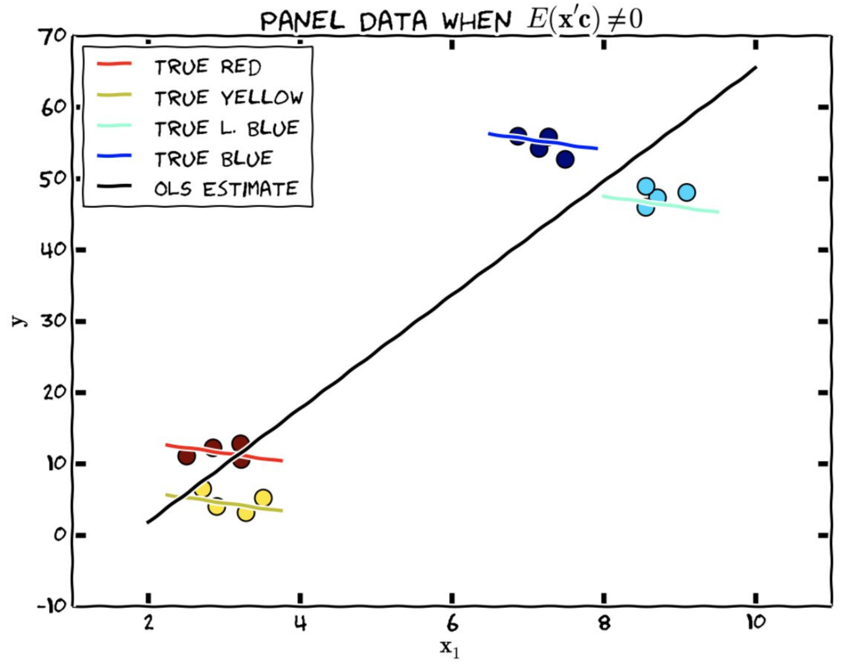
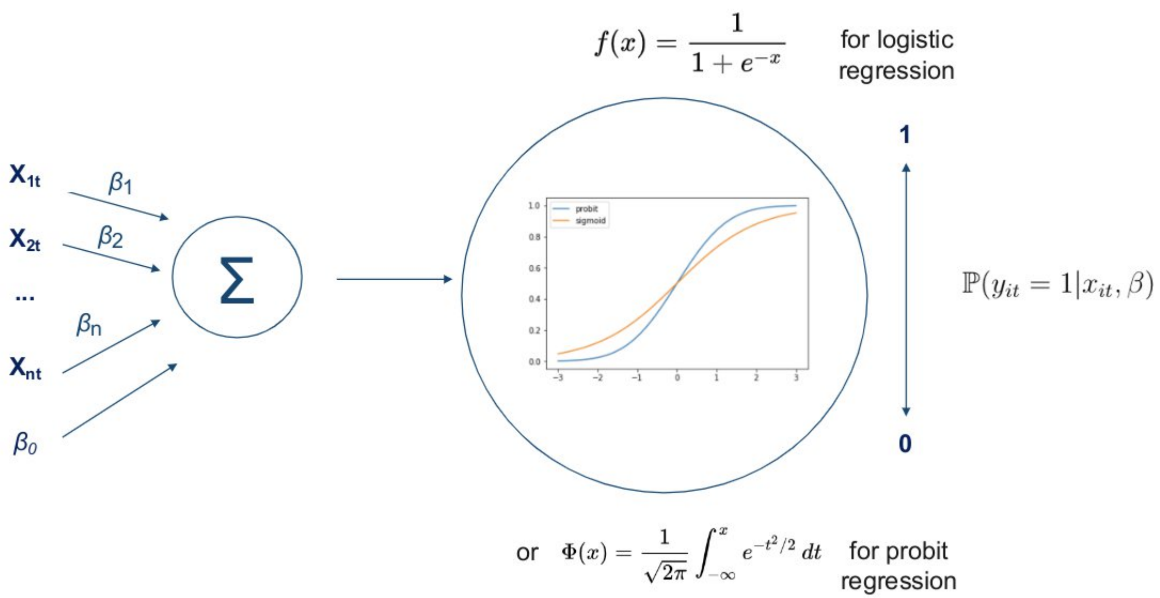



Figure 1. Unobserved Individual Heterogeneity and the Population Regression I

# Scalable package for logistic and probit regressions in TensorFlow does not exist

## Our solution: model as a single-layer neural network




First order approximation of the likelihood function with **mini-batch gradient descend methods**

# We build the first scalable statistical summary in TensorFlow

```
logitmodel=Pooled_logit_probit(link='logit')
logitmodel.fit(dataset, epochs=1, batch_size=128)
logitmodel.get_statistical_summary(robust=False, cofi_level=0.95)
```

```
1954/1954 [==============================] - 3s 2ms/step - loss: 0.5774
Deviance: 937583.4917471232
Null Deviance: 1386294.3575198718
p-value of LR test: 0.0
McFadden-R2: 0.3236764712622129
McFadden-R2 adj.: 0.323669979134512
AIC: 937605.4917471232
BIC: 937735.4623632608
Significance codes: 0. < *** < 0.001 <Текст< 0.01 < * < 0.05 < . < 0.1 <  < 1
```

| | coefficent | standard error | z_score | p-value | Confidence Interval | Significance |
|---|---|---|---|---|---|---|
| (intercept) | -0.198416 | 0.002538 | -78.182719 | 0.0 | [-0.20339, -0.193442] | *** |
| 1 | -0.036824 | 0.004025 | -9.148559 | 0.0 | [-0.044713, -0.028935] | *** |
| 2 | -0.849229 | 0.004149 | -204.695446 | 0.0 | [-0.85736, -0.841097] | *** |
| 3 | 1.175609 | 0.004254 | 276.380048 | 0.0 | [1.167272, 1.183946] | *** |
| 4 | -0.921350 | 0.004172 | -220.837964 | 0.0 | [-0.929527, -0.913172] | *** |

Now one can compute statistical summary **for billions of observations**

# Computation of Standard Errors is challenging

Under certain regularity conditions, asymptotically $\hat{\beta}_{ML} \sim \mathcal{N}(\beta^*, \mathcal{I}(\beta^*)^{-1})$
where $\beta^*$ is a vector of true coefficients, $\hat{\beta}_{ML}$ maximum likelihood estimate, covariance matrix $\mathcal{I}(\beta^*)^{-1}$

$$E\left[\hat{\beta}_{ML}\right] \approx \beta^*$$

$$Var\left[\hat{\beta}_{ML}\right] \approx \mathcal{I}(\beta^*)^{-1}$$

$$\Rightarrow \quad \hat{Var}\left[\hat{\beta}_{ML}\right] \approx \mathcal{I}(\hat{\beta}_{ML})^{-1} \qquad se(\hat{\beta}) = \sqrt{\hat{Var}\left[\hat{\beta}_{ML}\right]}$$

where $\mathcal{I} = -E\left[\ell''(\hat{\beta})\right] = -E\left[H\right]$ is the expected Fisher information.

*For the logit and probit regressions on observations* $\{x_i \in \mathbb{R}^p\}_{i=1}^n$

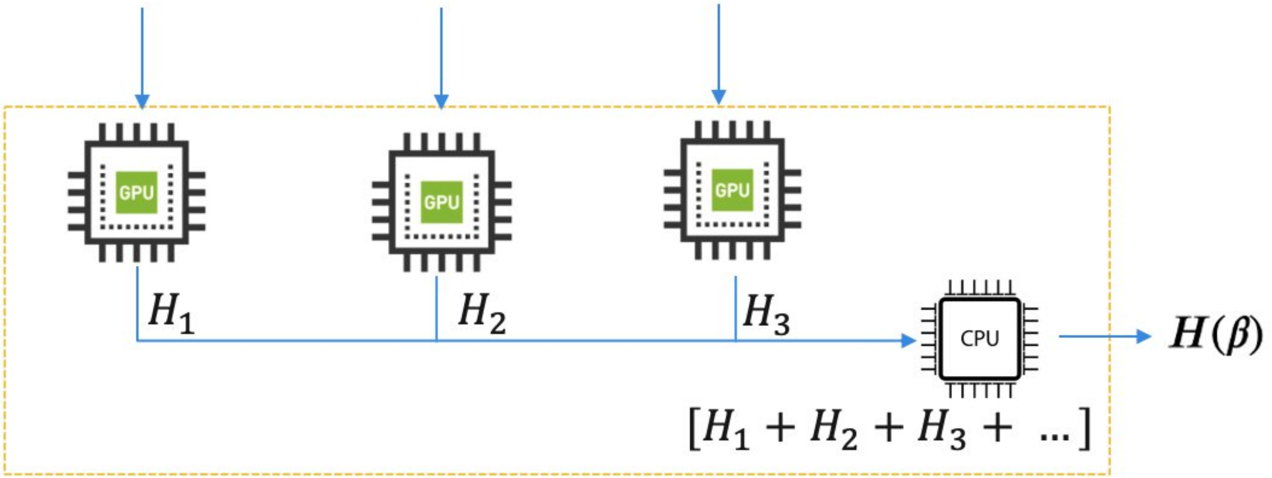
**Process the entire dataset!**

$$\mathcal{I}_{logit}(\beta) = \sum_{i=1}^{n} \frac{\exp(\beta^T x_i)}{(1+\exp(\beta^T x_i))^2} x_i x_i^T \qquad \mathcal{I}_{probit}(\beta) = \sum_{i=1}^{n} \frac{\phi(\beta^T x_i)^2}{\Phi(\beta^T x_i)(1-\Phi(\beta^T x_i))} x_i x_i^T$$

# Example of batchwise computations on multiple GPUs

***Example: Hessian Computation on an extra large dataset :*** $(X_i, y_i)_{i=1}^{N}$

$$H(\beta) = \sum_{i=1}^{N} \frac{\partial^2}{\partial\beta\partial\beta} l_i(\beta) = \sum_{i \in B_1} \frac{\partial^2}{\partial\beta\partial\beta} l_i(\beta) + \sum_{i \in B_2} \frac{\partial^2}{\partial\beta\partial\beta} l_i(\beta) + \sum_{i \in B_3} \frac{\partial^2}{\partial\beta\partial\beta} l_i(\beta) + \dots$$




**Entire summary is computed in similar scalable fashion including multiple GPU support**

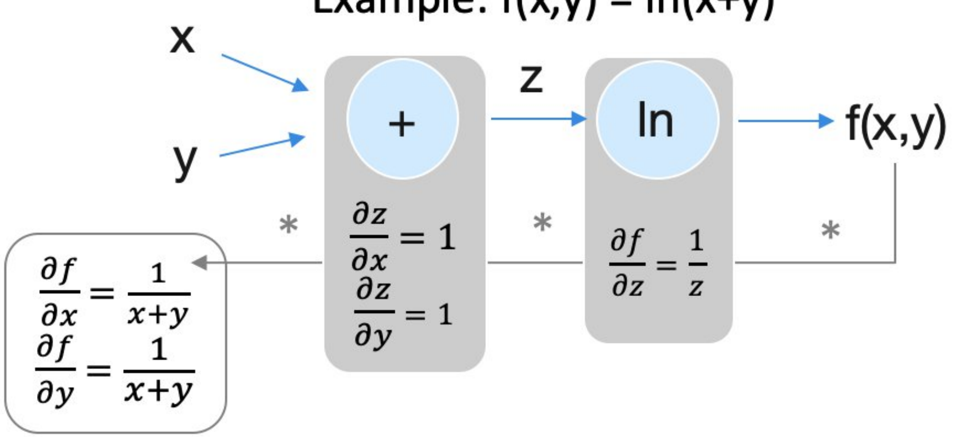
# Beyond Logit/Probit models in TensorFlow

***Logic should be the same…   BUT:  Traditional statistical software is rigid!***

***Compute SEs based on TFs AutoDiff :***

Use TensorFlows built-in **Automatic Differentiation** to compute gradients and hessians of log-likelihood necessary for SEs and robust sandwich errors

Example: f(x,y) = ln(x+y)



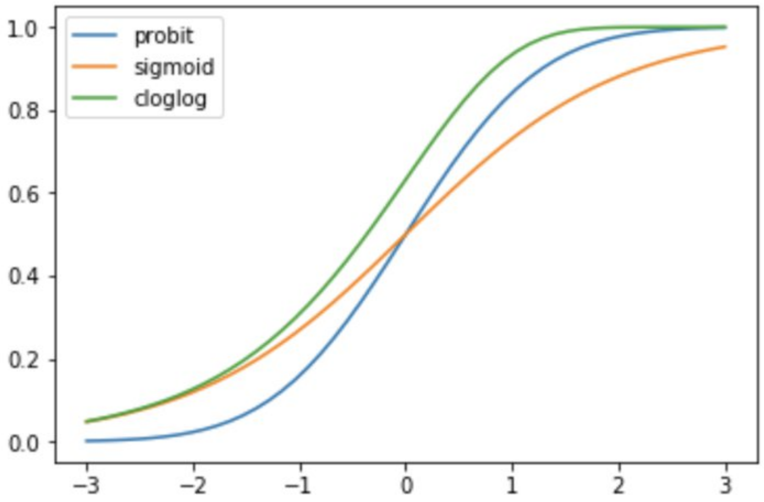

**Enable statistical inference for arbitrary model specification with same code!**

# Implementation of two use cases

***Choose arbitrary link for binary regression:***

```
def cloglog_link(x):
    y = 1-tf.math.exp(-tf.math.exp(x))
    return y

cloglogmodel=Pooled_logit_probit(link=cloglog_link)
```




***Simple Poisson Regression:***

```
def poisson_link(x):
    return tf.math.exp(x)

def neg_poissson_loglik(target, predicted):
    return -(target*tf.math.log(predicted)-predicted)

poissonmodel = Pooled_logit_probit(link = poisson_link, loss = neg_poisson_loglik)
```

# Pooled vs Fixed Effects Logit

$$\mathbb{P}(Y_{it} = 1 | x_{it}, \beta) = \sigma(x_{it}^T \beta + \beta_0)$$

$$\mathbb{P}(Y_{it} = 1 | x_{it}, \beta, \alpha_i) = \sigma(x_{it}^T \beta + \alpha_i)$$

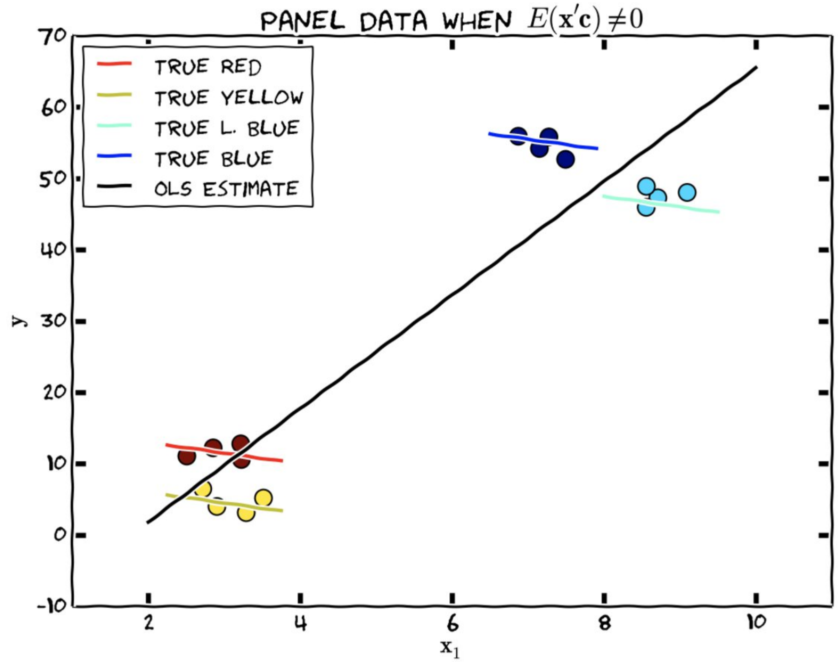



Figure 1. Unobserved Individual Heterogeneity and the Population Regression I

# Fixed Effect Model

## Why fixed effects?

- Catch the group-specific characteristics.

## Model setup $\mathbb{P}(Y_{it} = 1 \mid \beta_0, \alpha_{0i}, x_{it}) = \sigma(\alpha_{0i} + x_{it}\beta_0)$

1. $i \in \{1, 2, \ldots I\}$ denotes the group index.
2. $T_i$ denotes the number of observations in group $i$.
3. $\beta \in \mathbb{R}^R$ is the vector of coefficients.
4. $\alpha \in \mathbb{R}^I$ is the vector of fixed effects for each group.
5. $\mathbb{P}(Y_{it} = 1 \mid \beta, \alpha_i, x_{it}) = \sigma(\alpha_i + x_{it}\beta)$, where $\sigma(x) = \frac{e^x}{1+e^x}$.

Assume that

- $(Y_{it_1} \mid \beta)$ is independent of $(Y_{jt_2} \mid \beta)$ if $i \neq j$
- $(Y_{it_1} \mid \beta, \alpha_i)$ is independent of $(Y_{it_2} \mid \beta, \alpha_i)$ if $t_1 \neq t_2$.

# Parameter estimation

## *Unconditional MLE*

$$\prod_{i=1}^{I}\prod_{t=1}^{T_i}\sigma(\alpha_i + x_{it}\beta^T)^{y_{it}}(1-\sigma(\alpha_i + x_{it}\beta^T))^{1-y_{it}}$$

- Concavity of Loglikelihood function renders good performance using Newton's method
- Exploit sparsity of the Hessian, reducing computational complexity $O(N^3T^2)$ to $O(TN)$

# Incidental parameters problem

- Infinitely many parameters $\alpha_i$ with only $T_i$ observations result in inconsistency in $\alpha$ and bias in $\beta$.
- ***Bias correction method:***
  - Cross-over Jackknifing

$$\mathbb{E}[\hat{\beta}_{MLE}] = \beta_0 + \frac{B}{T} + o(\frac{1}{(T)^2}) \text{ for some constant B}$$

Bias doubled for half of the panel fit

$$\mathbb{E}[\hat{\beta}^1_{MLE}] = \mathbb{E}[\hat{\beta}^2_{MLE}] = \beta_0 + \frac{2B}{T} + o(\frac{1}{T^2})$$

So, Jackknife corrects the bias by a factor

$$\hat{\beta}^{JN}_{MLE} = 2\hat{\beta}_{MLE} - \frac{1}{2}(\hat{\beta}^1_{MLE} + \hat{\beta}^2_{MLE})$$

$$\mathbb{E}[\hat{\beta}^{JN}_{MLE}] = 2\mathbb{E}[\hat{\beta}_{MLE}] - \frac{1}{2}(\mathbb{E}[\hat{\beta}^1_{MLE}] + \mathbb{E}[\hat{\beta}^2_{MLE}]) = \beta_0 + o(\frac{1}{T^2})$$

# Conditional Fixed Effect Logit

- Why condition?
  - $\sum_{i=1}^{T_i} y_{it}$ sufficient for $\alpha_i$ so we maximize a condtional probabilty

  - analoguos to demeaning in linear fixed effects
  - removes incidental parameter problem -> consistent $\beta$ estimate

$$\mathbb{P}(Y_i = z \mid \sum_{t=1}^{T_i} Y_{it} = k, \beta, \alpha_i) = \frac{\mathbb{P}(Y_i = z \mid \beta, \alpha_i)}{\mathbb{P}(\sum_{t=1}^{T_i} Y_{it} = k \mid \beta, \alpha_i)}$$

$$\mathbb{P}(Y_i = (1,0,1)) = \frac{\mathbb{P}(Y_{i1}=1, Y_{i2}=0, Y_{i3}=1)}{\mathbb{P}(Y_{i1}+Y_{i2}+Y_{i3}=2)}$$

$$= \frac{exp\{(\beta^T(x_1+x_3)+2\alpha_i)\}}{exp\{(\beta^T(x_1+x_2)+2\alpha_i)\}+exp\{(\beta^T(x_1+x_3)+2\alpha_i)\}+exp\{(\beta^T(x_2+x_3)+2\alpha_i)\}}$$

# Conditional Logit

- Denominator term hard to calculate with $\binom{t_i}{T_i}$ terms $(t_i = \sum_{i=1}^{T_i} y_{it})$
  - Recursively O($T^2$)
  - Transfers to gradient and Hessian

- How do we get APEs ?
  - Conditional estimate of $\beta \rightarrow$ Unconditional MLE estimate $\alpha$, fixed $\beta$

- Comparison to Unconditional Logit
  - Unbiased, but computationally heavy O($T^2$)
  - (vs fast but biased O(T))

# The First Distributed Fixed Effects GLM !!

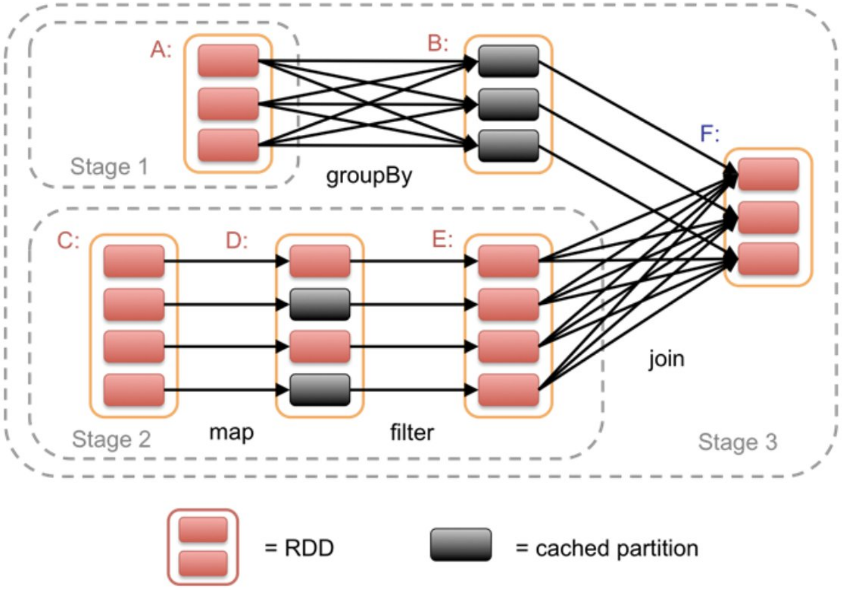
- Robust
  - Conditional and Unconditional MLE parameter estimation
  - Statistical summary (p-values, CI's, Wald and LL tests, Pseudo R^2)
  - Sandwich errors (no homoskedasticity assumption)
  - Incidental parameter and APE estimation
- Efficient
  - Local Numpy arrays
  - Vectorised implementation
  - BLAS for fast in-memory LA
- Stable
  - float64
  - Line search stabilization
  - Modified Cholesky decomposition (robust for collinearity)

# Our Implementation

## Spark-based Scalable Model

- The dataset is grouped by an observational unit
- One data shuffle
- RDD-s are cached on nodes
- Computation happens **only** with Transformations (Map) and Actions (Reduce)

**Log likelihood**
**Gradient**
**Hessian**
**APEs**
**Sandwich errors**



# Our Scalable Statistical summary

| Name | Parameter Estimate | Standard Error | z | p(>\|z\|) | * | Confidence Interval |
|---|---|---|---|---|---|---|
| | -1.786E-03 | 1.312E-03 | -1.362E+00 | 1.733E-01 | | [-4.357E-03, 7.846E-04] |
| | -2.319E+00 | 2.901E-01 | -7.995E+00 | 1.332E-15 | *** | [-2.888E+00, -1.751E+00] |
| | -5.809E-01 | 1.478E-01 | -3.930E+00 | 8.500E-05 | *** | [-8.706E-01, -2.912E-01] |
| | -1.167E+01 | 6.306E+01 | -1.851E-01 | 8.532E-01 | | [-1.353E+02, 1.119E+02] |
| | 4.027E-02 | 2.005E-02 | 2.009E+00 | 4.453E-02 | * | [9.848E-04, 7.956E-02] |
| | 3.828E-02 | 9.110E-02 | 4.202E-01 | 6.744E-01 | | [-1.403E-01, 2.168E-01] |
| | 5.462E-01 | 6.592E-02 | 8.286E+00 | 2.220E-16 | *** | [4.170E-01, 6.754E-01] |
| | -1.619E-03 | 2.763E-03 | -5.861E-01 | 5.578E-01 | | [-7.034E-03, 3.796E-03] |
| | 1.991E-01 | 1.107E-01 | 1.798E+00 | 7.215E-02 | | [-1.791E-02, 4.161E-01] |
| | -2.760E-01 | 8.948E-02 | -3.085E+00 | 2.036E-03 | ** | [-4.514E-01, -1.007E-01] |
| | -1.127E-03 | 1.360E-02 | -8.292E-02 | 9.339E-01 | | [-2.777E-02, 2.552E-02] |
| | -3.739E-09 | 9.172E-09 | -4.077E-01 | 6.835E-01 | | [-2.172E-08, 1.424E-08] |
| | 9.617E-01 | 5.652E-02 | 1.702E+01 | 0.000E+00 | *** | [8.510E-01, 1.073E+00] |
| | 2.213E-01 | 1.966E-02 | 1.126E+01 | 0.000E+00 | *** | [1.827E-01, 2.598E-01] |
| | 2.152E-05 | 2.000E-06 | 1.076E+01 | 0.000E+00 | *** | [1.760E-05, 2.544E-05] |
| | 6.162E-01 | 7.081E-02 | 8.702E+00 | 0.000E+00 | *** | [4.774E-01, 7.550E-01] |
| | 2.560E-02 | 5.183E-02 | 4.939E-01 | 6.214E-01 | | [-7.599E-02, 1.272E-01] |
| | -1.792E-01 | 7.471E-02 | -2.398E+00 | 1.647E-02 | * | [-3.256E-01, -3.275E-02] |
| | -1.371E-01 | 7.314E-02 | -1.875E+00 | 6.076E-02 | | [-2.805E-01, 6.198E-03] |
| | 1.572E-03 | 3.599E-04 | 4.367E+00 | 1.261E-05 | *** | [8.662E-04, 2.277E-03] |
| | 2.074E-01 | 9.095E-02 | 2.280E+00 | 2.259E-02 | * | [2.913E-02, 3.857E-01] |
| | -3.370E-02 | 4.761E-03 | -7.078E+00 | 1.461E-12 | *** | [-4.303E-02, -2.437E-02] |
| | -5.819E-02 | 1.828E-01 | -3.184E-01 | 7.502E-01 | | [-4.164E-01, 3.001E-01] |
| | 2.163E-01 | 9.214E-02 | 2.348E+00 | 1.889E-02 | * | [3.572E-02, 3.969E-01] |

LR-test: 1.897E+03 p-value: 0.000E+00

Pseudo R^2: 0.13085

AIC: 12643.53950

# Results

We can perform statistical inference on **BIG DATA**!

- In TensorFlow, Pooled regressions with arbitrary link functions
- In Apache Spark, scalable linear models with Unobserved effects

| Coefficient | APE_logit | PE_at_mean_logit | APE_probit | PE_at_mean_probit | APE_uncond |
|---|---|---|---|---|---|
| **0** | -5.88E-10 | -5.54E-10 | -5.32E-10 | -5.29E-10 | 0.00E+00 |
| **1** | 5.39E-04 | 5.08E-04 | 5.78E-04 | 5.75E-04 | 0.00E+00 |
| **2** | -1.42E-04 | -1.34E-04 | -1.47E-04 | -1.47E-04 | -1.45E-04 |
| **3** | 7.69E-04 | 7.25E-04 | 8.07E-04 | 8.03E-04 | 2.77E-04 |
| **4** | 1.49E-06 | 1.40E-06 | 1.54E-06 | 1.53E-06 | 0.00E+00 |
| **5** | 2.57E-02 | 2.57E-02 | 2.46E-02 | 2.46E-02 | 0.00E+00 |

Figure 10: Average Partial Effects.