Classification under Nuisance Parameters and Generalized Label Shift in Likelihood-Free Inference

Rafael Izbicki
(UFSCar)

Acknowledgments: FAPESP, CAPES, CNPq

L. Masserano
(Stats&DS/CMU)

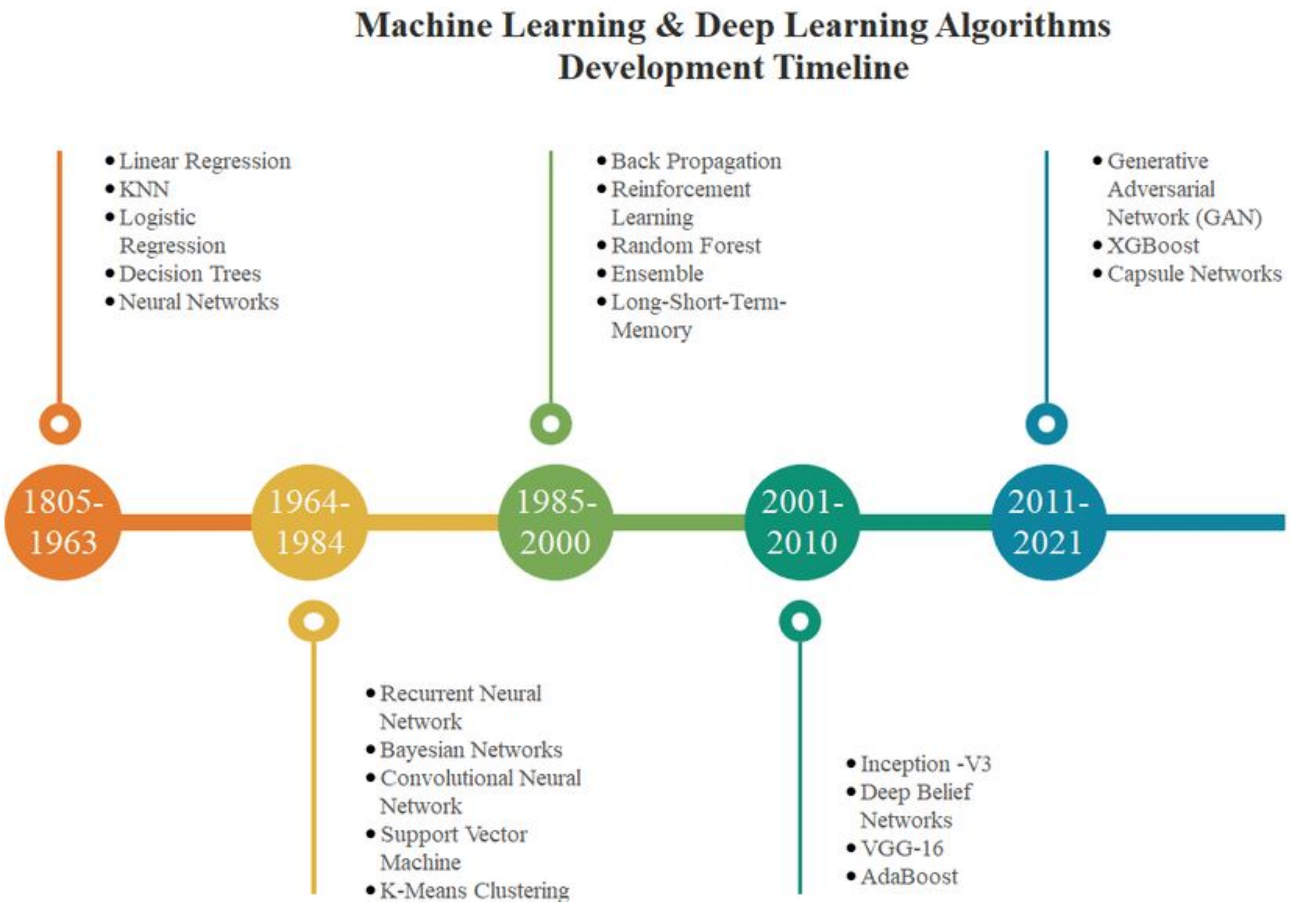
A. Shen
(Stats&DS/CMU)

T. Dorigo (INFN)

M. Doro (Padova)

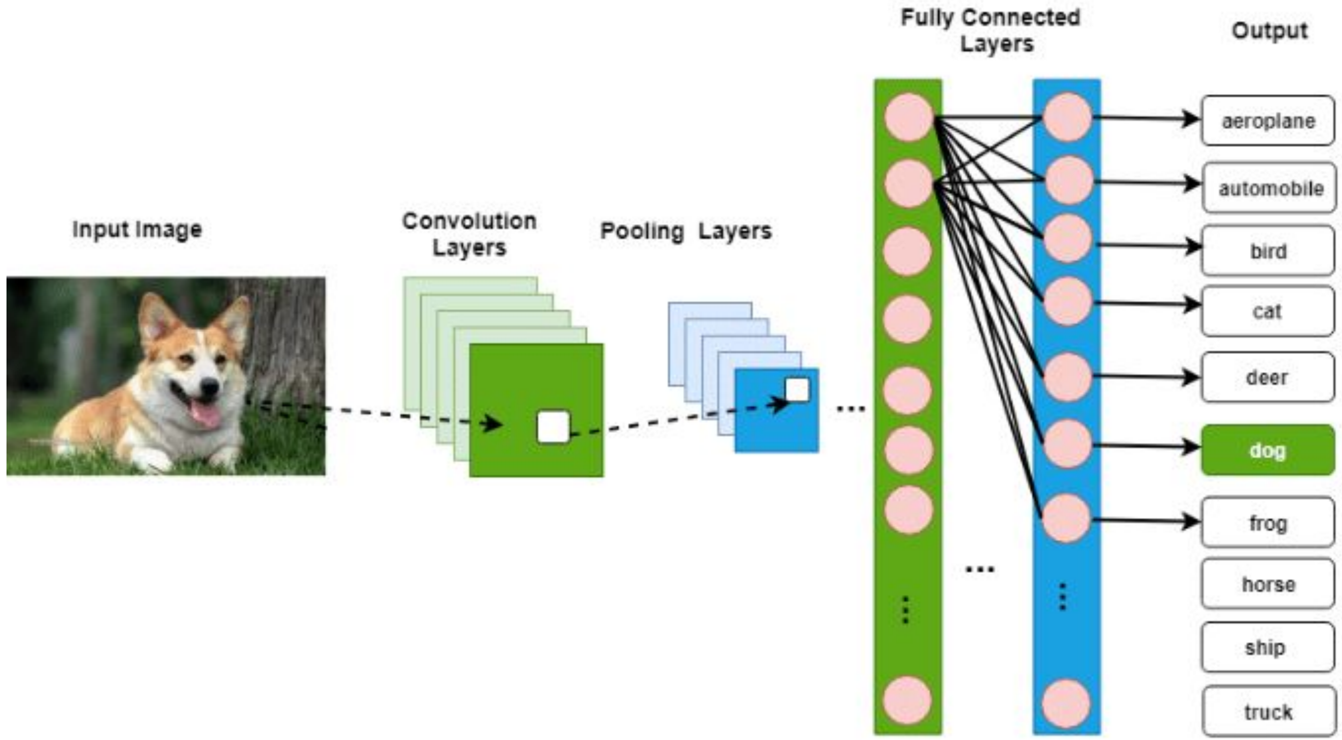
A. B. Lee (Stats&DS/CMU)

Acknowledgments: CNPq, FAPESP, NSF



Source: https://www.mdpi.com/2227-9032/10/3/541

Machine Learning Revolution



Goal in Supervised Learning

Given measurements $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$, learn a model to predict Y_i based on $\mathbf{X}_i$

construct g to obtain good predictions

$$g(\mathbf{X}_{n+1}) \approx Y_{n+1}, \ \ldots, \ g(\mathbf{X}_{n+m}) \approx Y_{n+m}$$

$$R(g) = \mathbb{E}\left[(Y - g(\mathbf{X}))^2\right]$$

Standard Assumption: i.i.d.

If data is not i.i.d., new assumptions are necessary

Information Sciences 649 (2023) 119612

ELSEV

A unif

Felipe M
Juan Pa

[a] Department of Statistics, University of Michigan, United States of America
[b] Department of Statistics, Federal University of São Carlos, Brazil
[c] Trustly, Brazil
[d] Experian LatAm DataLab, Brazil
[e] Department of Applied Mathematics, Institute of Mathematics and Statistics, University of São Paulo, Brazil

- **[Total Dataset Shift]** $H_{0,\mathrm{D}} : P^{(1)}_{\mathbf{X},Y} = P^{(2)}_{\mathbf{X},Y}$
- **[Feature Shift]** $H_{0,\mathrm{F}} : P^{(1)}_{\mathbf{X}} = P^{(2)}_{\mathbf{X}}$
- **[Response Shift]** $H_{0,\mathrm{R}} : P^{(1)}_{Y} = P^{(2)}_{Y}$
- **[Conditional Shift - Type 1]** $H_{0,\mathrm{C1}} : P^{(1)}_{\mathbf{X}|Y} = P^{(2)}_{\mathbf{X}|Y} \quad (P^{(2)}_{Y}\text{- almost surely})$
- **[Conditional Shift - Type 2]** $H_{0,\mathrm{C2}} : P^{(1)}_{Y|\mathbf{X}} = P^{(2)}_{Y|\mathbf{X}} \quad (P^{(2)}_{\mathbf{X}}\text{- almost surely})$

Example: Y|x is the same

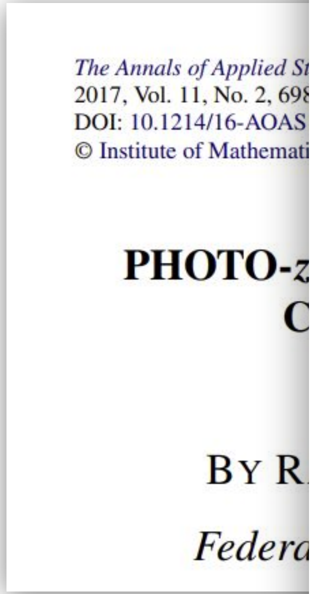
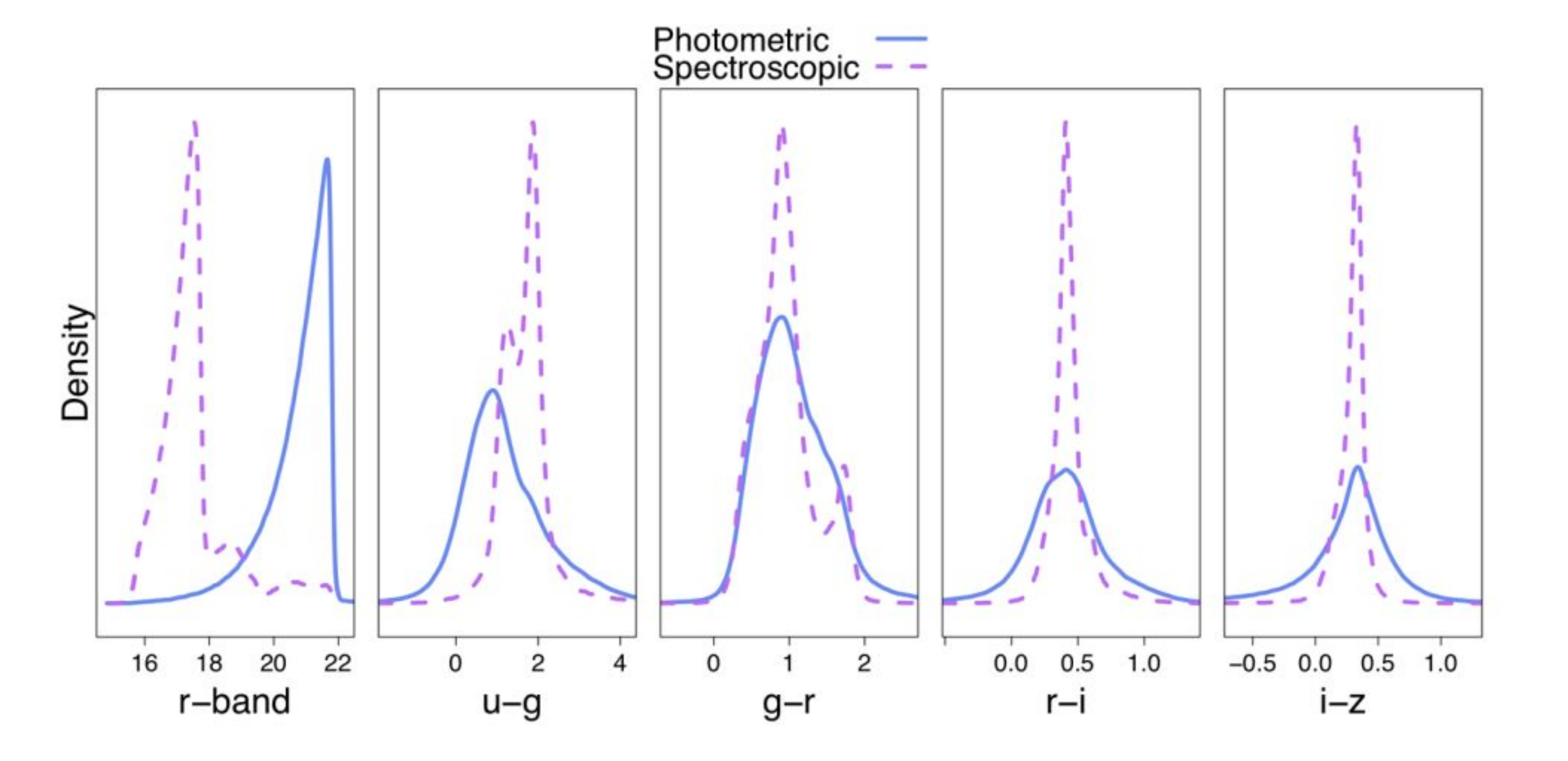

The Annals of Applied St
2017, Vol. 11, No. 2, 698
DOI: 10.1214/16-AOAS
© Institute of Mathemati

PHOTO-z
C

By R

Federa



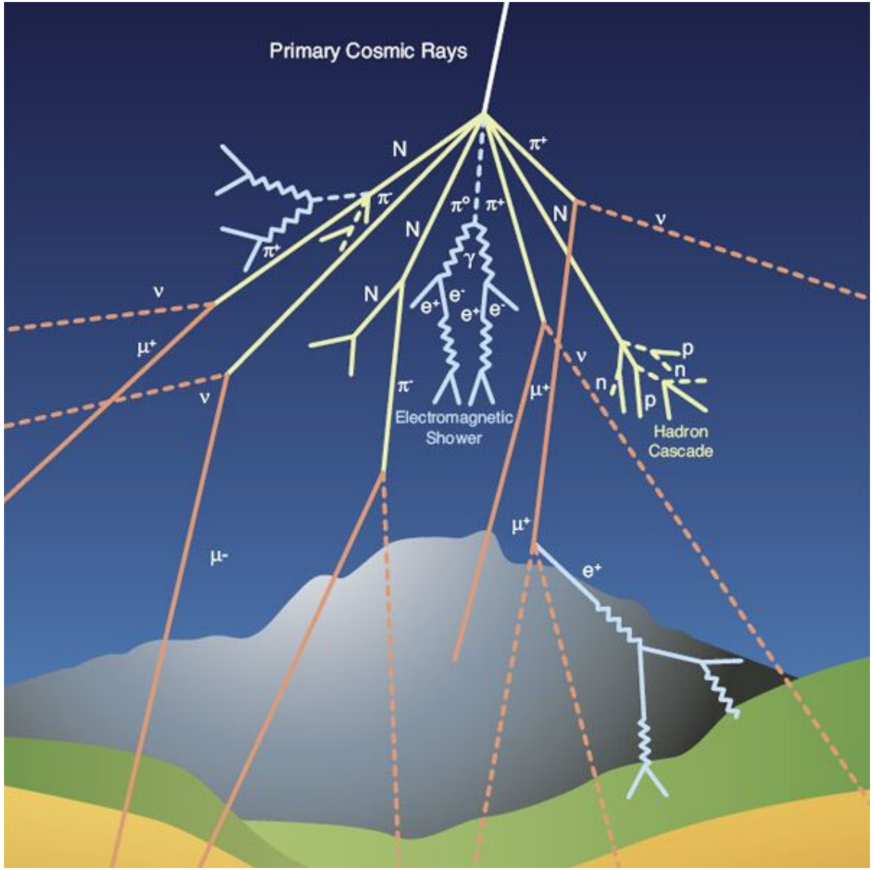
Example: X|y is the same

Journal of Machine Learning Research 20 (2019) 1-33 Submitted 7/18; Revised 2/19; Published 4/19

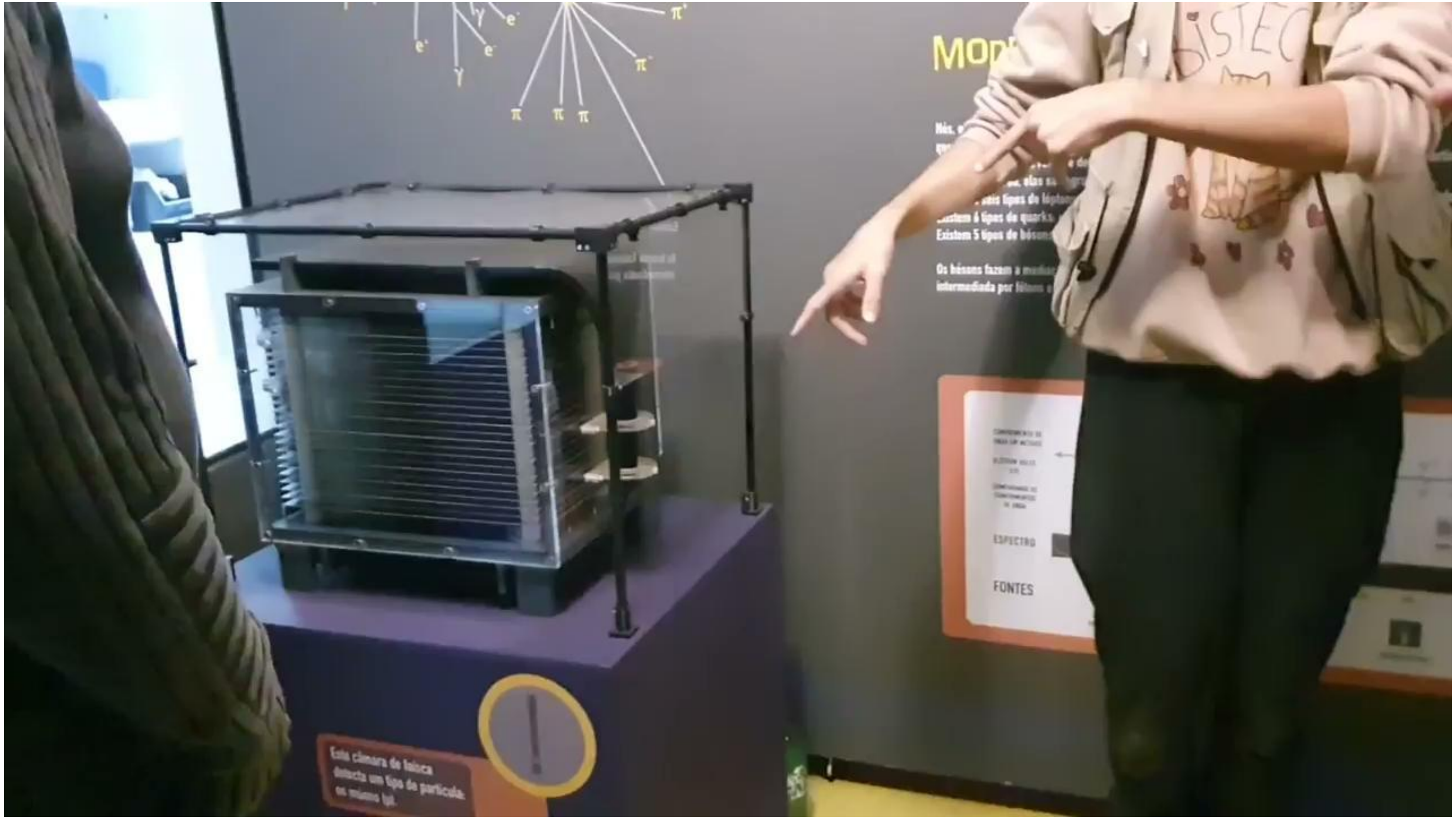
Quantification Under Prior Probability Shift: the Ratio Estimator and its Extensions

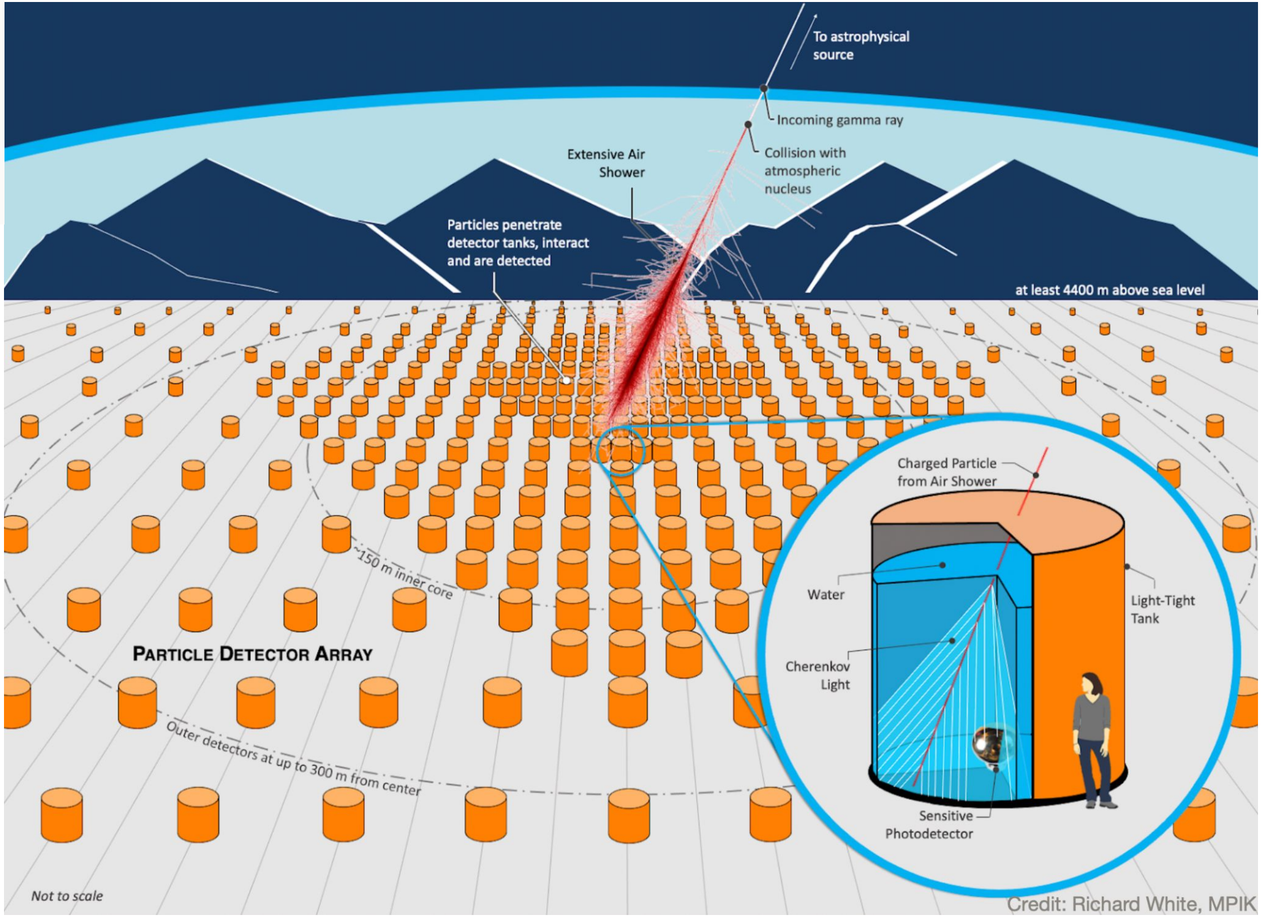
Afonso Fernandes Vaz AFONSOFVAZ@GMAIL.COM
Rafael Izbicki RAFAELIZBICKI@GMAIL.COM
Rafael Bassi Stern RBSTERN@GMAIL.COM
Department of Statistics
Federal University of São Carlos
São Carlos, SP 13565-905, Brazil

This work



Primary Cosmic Rays
N
π+
π-
π+
π°
π+
N
N
ν
γ
N
e-
e+
e+
e-
μ+
ν
p
n
ν
n
π-
μ+
p
Electromagnetic Shower
Hadron Cascade
μ+
μ-
e+


ESPECTRO
FONTES
Esta câmara de faísca
detecta um tipo de partícula:

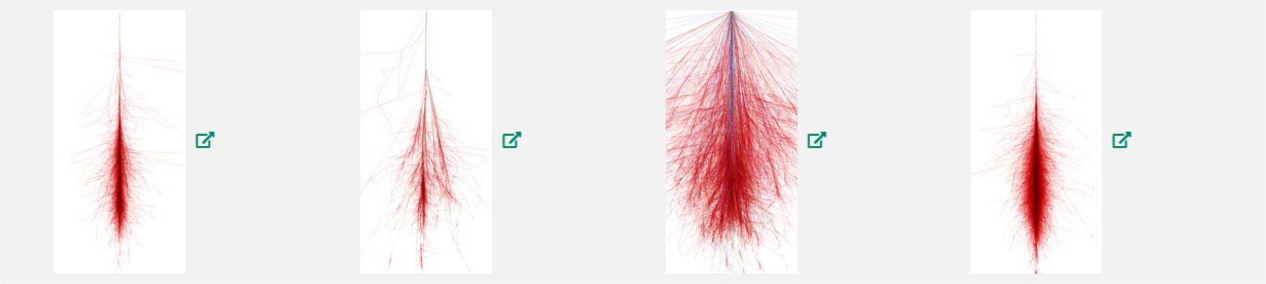


To astrophysical source
Incoming gamma ray
Collision with atmospheric nucleus
Extensive Air Shower
Particles penetrate detector tanks, interact and are detected
at least 4400 m above sea level
~150 m inner core
PARTICLE DETECTOR ARRAY
Outer detectors at up to 300 m from center
Charged Particle from Air Shower
Water
Light-Tight Tank
Cherenkov Light
Sensitive Photodetector
Not to scale
Credit: Richard White, MPIK

Home Legals Privacy Policy Sitemap KIT

CORSIKA Philosophy Documentation Downloads CORSIKA 8 Development

CORSIKA – an Air Shower Simulation Program



Photon shower Proton shower Iron shower Muon shower

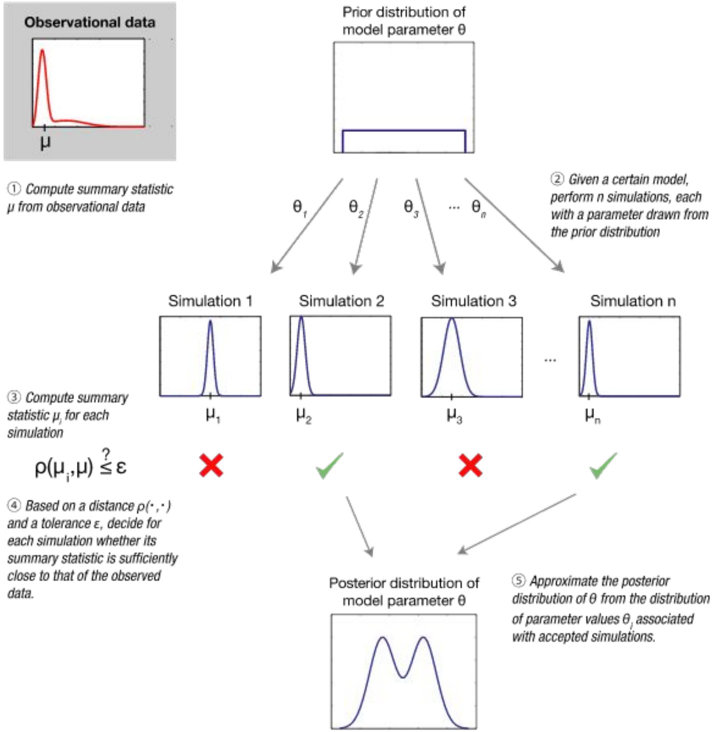
(Compiled by Fabian Schmidt, University of Leeds, UK)

In terms of Statistics:

$$\text{mechanistic model:} \quad \boldsymbol{\theta} = (Y, \boldsymbol{\nu}) \mapsto \mathbf{X}.$$

$$\{(Y_i, \mathbf{x}_i)\}_{i=1}^{B}$$

Likelihood-Free Inference (Simulator-Based Inference)

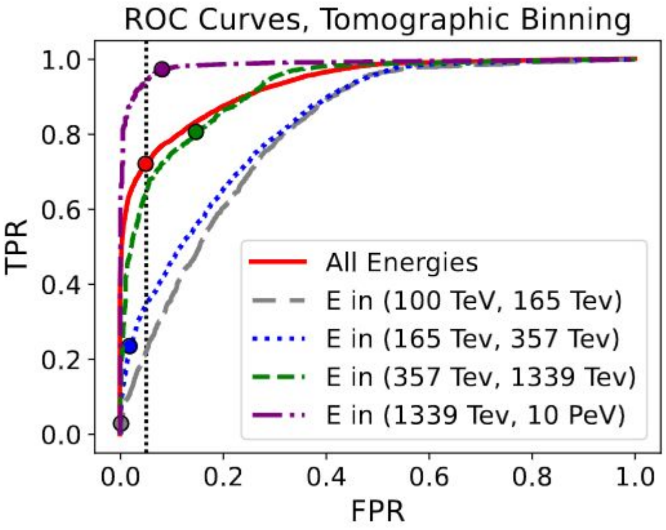


Source: wikipedia

P(Y=1|x) leads to invalid uncertainty quantification

$$\{(Y_i, \mathbf{x}_i)\}_{i=1}^{B}$$

ROC Curves are not valid



Summarizing so far

$$\text{mechanistic model:} \quad \boldsymbol{\theta} = (Y, \boldsymbol{\nu}) \mapsto \mathbf{X}.$$

$$\{(Y_i, \mathbf{x}_i)\}_{i=1}^{B}$$

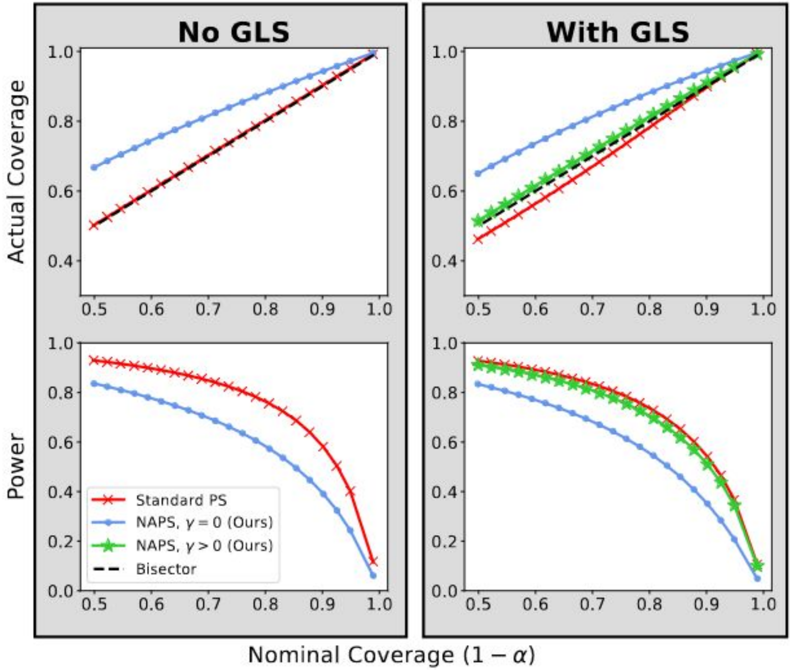
P(Y=1|x): invalid uncertainty quantification

$$\mathbb{P}_{\text{target}}(Y \in R_\alpha(\mathbf{X})) \geq 1 - \alpha$$

Generalized Label Shift (GLS)

$$p_{\text{train}}(\mathbf{x}|y, \boldsymbol{\nu}) = p_{\text{target}}(\mathbf{x}|y, \boldsymbol{\nu})$$

$$\mathbb{P}_{\text{target}}(Y \in R_\alpha(\mathbf{X}))$$



Method

$$H_{0,y} : Y = y \text{ versus } H_{1,y} : Y \neq y.$$

$$\lambda(\mathbf{x}) = \mathbb{P}_{\text{train}}(Y = y|\mathbf{x})$$

$$W_\lambda(C; y, \boldsymbol{\nu}) := \mathbb{P}_{target}(\lambda(\mathbf{X}) \leq C|y, \boldsymbol{\nu})$$

$$p_{\text{train}}(\mathbf{x}|y, \boldsymbol{\nu}) = p_{\text{target}}(\mathbf{x}|y, \boldsymbol{\nu})$$

Monotonic Classifier

$$W_\lambda(C; y, \boldsymbol{\nu}) := \mathbb{P}_{target}\left(\lambda(\mathbf{X}) \leq C | y, \boldsymbol{\nu}\right)$$

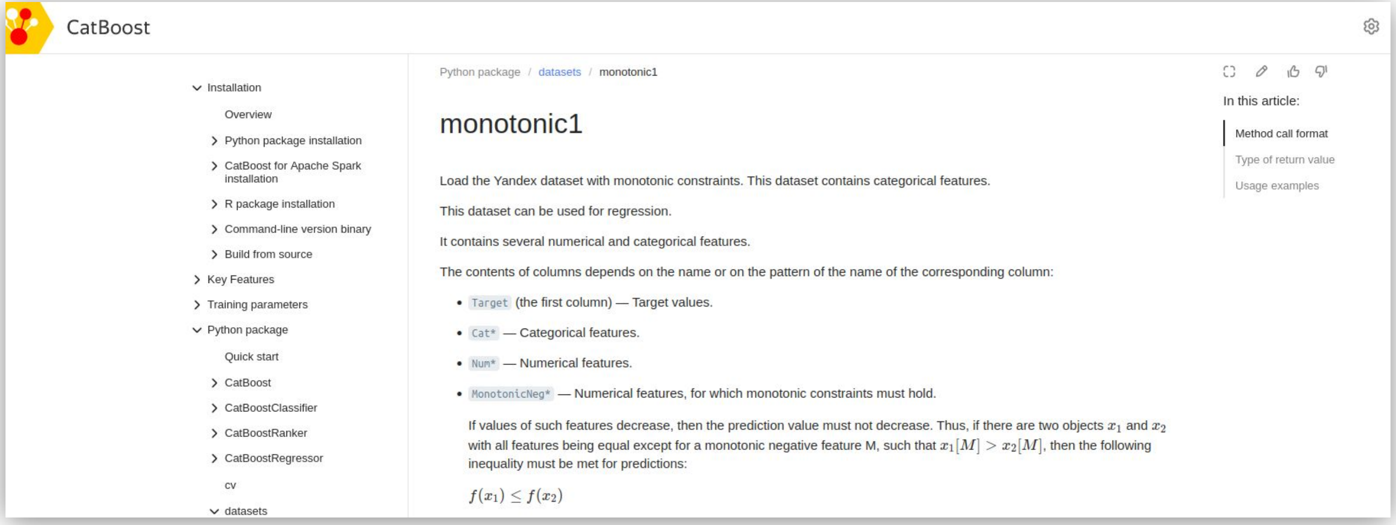
We learn $W_\lambda(C; y, \boldsymbol{\nu})$ using a monotone regression that enforces that the rejection probability is a non-decreasing function of C. For each point i ($i = 1, \ldots, B$) in the calibration sample $\mathcal{T}' = \{(y_1, \boldsymbol{\nu}_1, \mathbf{X}_1), \ldots, (y_B, \nu_B, \mathbf{X}_B)\}$ sampled from $p_{\text{train}}(\boldsymbol{\theta})\mathcal{L}(\mathbf{x}; \boldsymbol{\theta})$, with $\boldsymbol{\theta} = (y, \boldsymbol{\nu})$, we sample a set of K cutoffs from the empirical distribution of the test statistic λ. Then, we regress the random variable

$$Z_{i,j} := \mathbb{I}\left(\lambda(\mathbf{X}_i) \leq C_j\right) \tag{13}$$

on y_i, $\boldsymbol{\nu}_i$ and $C_{i,j}$ ($= C_j$) using the "augmented" calibration sample $\mathcal{T}'' = \{(y_i, \boldsymbol{\nu}_i, C_{i,j}, Z_{i,j})\}_{i,j}$, for $i = 1, \ldots, B$ and $j = 1, \ldots, K$, where K is our augmentation factor. See Algorithm 1 for details.

Monotonic Classifier

$$W_\lambda(C; y, \boldsymbol{\nu}) := \mathbb{P}_{target}\left(\lambda(\mathbf{X}) \leq C | y, \boldsymbol{\nu}\right)$$


CatBoost

- Installation
  - Overview
  - Python package installation
  - CatBoost for Apache Spark installation
  - R package installation
  - Command-line version binary
  - Build from source
- Key Features
- Training parameters
- Python package
  - Quick start
  - CatBoost
  - CatBoostClassifier
  - CatBoostRanker
  - CatBoostRegressor
  - cv
  - datasets

Python package / datasets / monotonic1

monotonic1

Load the Yandex dataset with monotonic constraints. This dataset contains categorical features.

This dataset can be used for regression.

It contains several numerical and categorical features.

The contents of columns depends on the name or on the pattern of the name of the corresponding column:

- `Target` (the first column) — Target values.
- `Cat*` — Categorical features.
- `Num*` — Numerical features.
- `MonotonicNeg*` — Numerical features, for which monotonic constraints must hold.

  If values of such features decrease, then the prediction value must not decrease. Thus, if there are two objects x_1 and x_2 with all features being equal except for a monotonic negative feature M, such that $x_1[M] > x_2[M]$, then the following inequality must be met for predictions:

  $f(x_1) \leq f(x_2)$

In this article:

Method call format

Type of return value

Usage examples

Controlling FPR or TPR

α

$$\texttt{FPR}(C;\boldsymbol{\nu}) := W_{\widehat{\lambda}}(C;0,\boldsymbol{\nu})$$

$$\texttt{TPR}(C;\boldsymbol{\nu}) := W_{\widehat{\lambda}}(C;1,\boldsymbol{\nu})$$

1-β

$$C_\alpha = \inf_{\boldsymbol{\nu}\in\mathcal{N}} \texttt{FPR}^{-1}(\alpha;\boldsymbol{\nu}),$$

$$\widetilde{C}_\alpha = \sup_{\boldsymbol{\nu}\in\mathcal{N}} \texttt{TPR}^{-1}(\alpha;\boldsymbol{\nu})$$

Controlling FPR or TPR, but with more power

$$C^*_\alpha(\mathbf{x}) = \inf_{\boldsymbol{\nu} \in S(\mathbf{x};\gamma)} \{\mathrm{FPR}^{-1}(\beta; \boldsymbol{\nu})\}$$

$$\beta = \alpha - \gamma$$

arXiv > stat > arXiv:2107.03920

Search...

Help | Adv

Statistics > Machine Learning

[Submitted on 8 Jul 2021 (v1), last revised 19 Nov 2023 (this version, v8)]

Likelihood-Free Frequentist Inference: Bridging Classical Statistics and Machine Learning for Reliable Simulator-Based Inference

Niccolò Dalmasso, Luca Masserano, David Zhao, Rafael Izbicki, Ann B. Lee

Many areas of science make extensive use of computer simulators that implicitly encode intractable likelihood functions of complex systems. Classical statistical methods are poorly suited for these so-called likelihood-free inference (LFI) settings, especially outside asymptotic and low-dimensional regimes. At the same time, traditional LFI methods - such as Approximate Bayesian Computation or more recent machine learning techniques - do not guarantee confidence sets with nominal coverage in general settings (i.e., with high-dimensional data, finite sample sizes, and for any parameter value). In addition, there are no diagnostic tools to check the empirical coverage of confidence sets provided by such methods across the entire parameter space. In this work, we propose a unified and modular inference framework that bridges classical statistics and modern machine learning providing (i) a practical approach to the Neyman construction of confidence sets with frequentist finite-sample coverage for any value of the unknown parameters; and (ii) interpretable diagnostics that estimate the empirical coverage across the entire parameter space. We refer to the general framework as likelihood-free frequentist inference (LF2I). Any method that defines a test statistic can leverage LF2I to create valid confidence sets and diagnostics without costly Monte Carlo samples at fixed parameter settings. We study the power of two likelihood-based test statistics (ACORE and BFF) and demonstrate their empirical performance on high-dimensional, complex data. Code is available at this https URL.

Controlling FPR or TPR, but with more power

Theorem 1 (Nuisance-aware cutoffs for FPR/TPR control). *Choose a threshold* $\alpha \in [0, 1]$ *and* $\gamma \in [0, \alpha]$. *Let* $S_y(\mathbf{x}; \gamma)$ *be a valid* $(1 - \gamma)$ *confidence set for* $\boldsymbol{\nu}$ *at fixed* $y \in \{0, 1\}$ *according to Definition 3. Let* $\lambda(\mathbf{X})$ *be any test statistic that measures how plausible it is that* $\mathbf{X}$ *was generated from* $H_{0,y}$. *Define the nuisance-aware rejection cutoff to be*

$$C^*_{\alpha,y}(\mathbf{x}) = \inf_{\boldsymbol{\nu} \in S_y(\mathbf{x};\gamma)} \{W_\lambda^{-1}(\beta; y, \boldsymbol{\nu})\}, \tag{8}$$

where $\beta = \alpha - \gamma$, *and* W *is the rejection probability in Definition 1. Then, for all* $\boldsymbol{\nu} \in \mathcal{N}$, *we have FPR control:*

$$\mathbb{P}_{target}\left(\lambda(\mathbf{X}) \leq C^*_{\alpha,y}(\mathbf{X}) | y, \boldsymbol{\nu}\right) \leq \alpha \tag{9}$$

(maximum type-I error probability for $H_{0,y}$*).*

NAPS

Definition 2 (Nuisance-aware prediction set). *A nuisance-aware prediction set (NAPS) is the set returned from a set-valued classifier* $\mathbf{H} : \mathbf{x} \mapsto \{\emptyset, 0, 1, \{0,1\}\}$ *with*

$$\mathbf{H}(\mathbf{x}; \alpha) = \left\{ y \in \{0,1\} \mid \widehat{\tau}_y(\mathbf{x}) > C^*_{\alpha,y}(\mathbf{x}) \right\}, \qquad (5)$$

where

$$C^*_{\alpha,y}(\mathbf{x}) = \inf_{\boldsymbol{\nu} \in S_y(\mathbf{x};\gamma)} \{ W^{-1}_{\widehat{\tau}_y}(\beta; y, \boldsymbol{\nu}) \}, \qquad (6)$$

is the rejection cutoff, $\beta = \alpha - \gamma$ *and* $S_y(\mathbf{x};\gamma)$ *is a* $(1-\gamma)$ *confidence set for* $\boldsymbol{\nu}$ *defined by Equation 7.*

NAPS

Theorem 2. *Let* $\mathbf{H}(\mathbf{x};\alpha)$ *be the nuisance-aware prediction set of Definition* 2. *Under GLS, for every* $y \in \{0,1\}$ *and* $\nu \in \mathcal{N}$

$$\mathbb{P}_{target}(Y \in \mathbf{H}(\mathbf{X};\alpha)|y,\nu) \geq 1-\alpha.$$

Moreover,

$$\mathbb{P}_{target}(Y \in \mathbf{H}(\mathbf{X};\alpha)) \geq 1-\alpha.$$

Summary

Algorithm 1 `Nuisance-aware prediction sets`

Input: training set $\mathcal{T} = \{(Y_i, \mathbf{X}_i)\}_{i=1}^{B}$; calibration set $\mathcal{T}' = \{(y_i, \boldsymbol{\nu}_i, \mathbf{X}_i)\}_{i=1}^{B'}$; observation $\mathbf{x}$; miscoverage levels $\alpha \in (0,1)$ and $\gamma \in [0, \alpha)$.

Output: Prediction set $H_\alpha(\mathbf{x})$ such that Equation 1 holds.

// Training

Estimate $\mathbb{P}_{\text{train}}(Y = y|\mathbf{X})$ with a probabilistic classifier

// Calibration

Estimate $W_{\tau_y}(C; y, \boldsymbol{\nu}) := \mathbb{P}_{\text{target}}(\tau_y(\mathbf{X}) \leq C | y, \boldsymbol{\nu})$ as detailed in Algorithm 2 by

i. Computing $\widehat{\tau}_y(\mathbf{X})$ as in Equation 3 for all $\mathbf{X} \in \mathcal{T}'$;

ii. Constructing the augmented calibration set $\mathcal{T}''$;

iii. Estimating $W_{\tau_y}(C; y, \boldsymbol{\nu})$ from $\mathcal{T}''$ via monotone regression.

// Inference

for $y \in \{0, 1\}$ **do**

  Compute $\widehat{\tau}_y(\mathbf{x})$ as in Equation 3

  if $\gamma = 0$ **then**

    $C^*_{\alpha,y}(\mathbf{x}) \leftarrow \inf_{\boldsymbol{\nu} \in \mathcal{N}} \{\widehat{W}^{-1}_{\widehat{\tau}_y}(\alpha; y, \boldsymbol{\nu})\}$

  else

    Obtain a level-γ confidence set $S_y(\mathbf{x}; \gamma)$ for $\boldsymbol{\nu}$

    $C^*_{\alpha,y}(\mathbf{x}) \leftarrow \inf_{\boldsymbol{\nu} \in S_y(\mathbf{x};\gamma)} \{\widehat{W}^{-1}_{\widehat{\tau}_y}(\alpha - \gamma; y, \boldsymbol{\nu})\}$

  end if

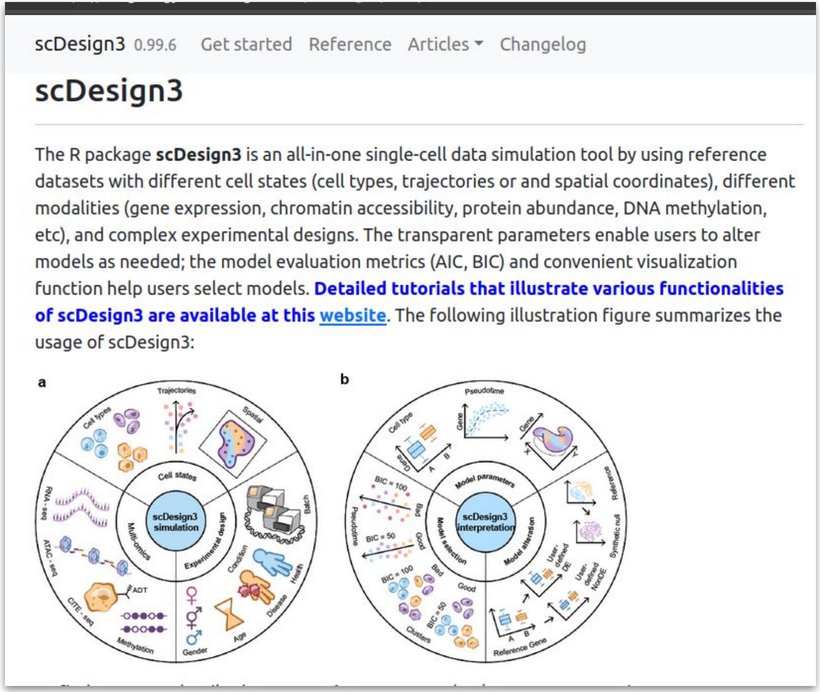
end for

$\mathbf{H}(\mathbf{x}; \alpha) \leftarrow \{y \in \{0,1\} \mid \widehat{\tau}_y(\mathbf{x}) > C^*_{\alpha,y}(\mathbf{x})\}$

return $\mathbf{H}(\mathbf{x}; \alpha)$

Application: Single-Cell RNA Sequencing

Goal: infer the cell's type (CD4+ vs Cytotoxic T-cells) from the observed gene count



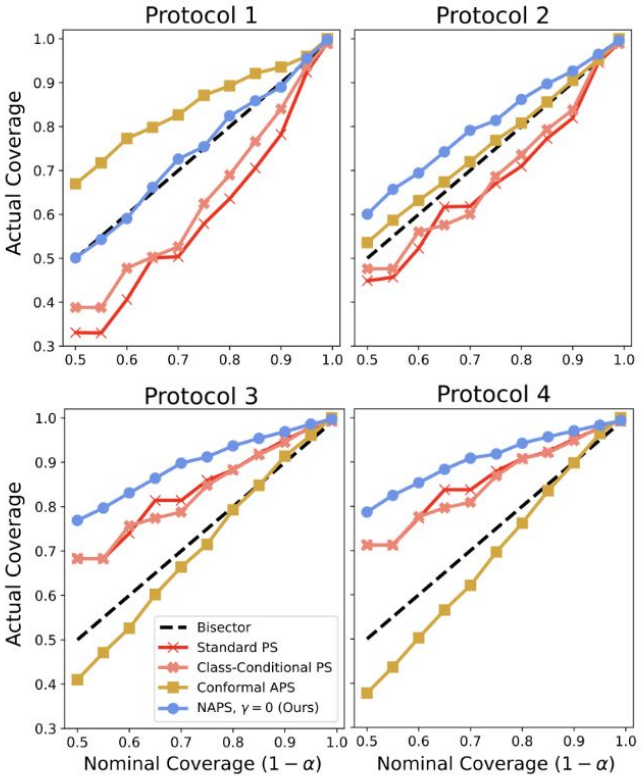
scDesign3 0.99.6 Get started Reference Articles Changelog

scDesign3

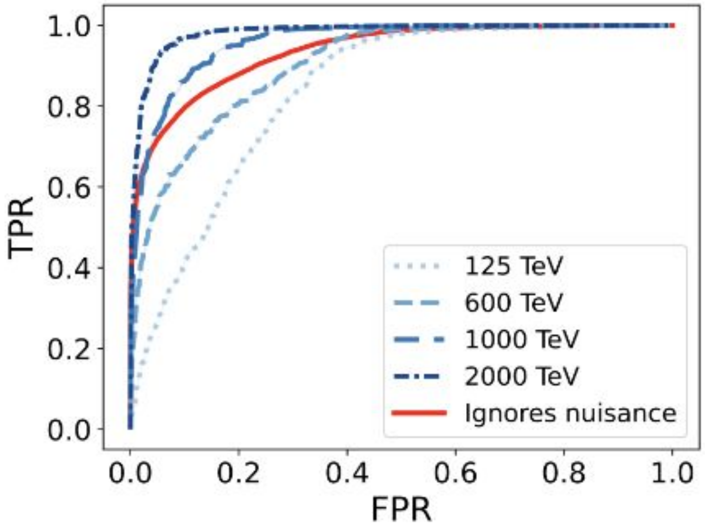
The R package **scDesign3** is an all-in-one single-cell data simulation tool by using reference datasets with different cell states (cell types, trajectories or and spatial coordinates), different modalities (gene expression, chromatin accessibility, protein abundance, DNA methylation, etc), and complex experimental designs. The transparent parameters enable users to alter models as needed; the model evaluation metrics (AIC, BIC) and convenient visualization function help users select models. **Detailed tutorials that illustrate various functionalities of scDesign3 are available at this website**. The following illustration figure summarizes the usage of scDesign3:

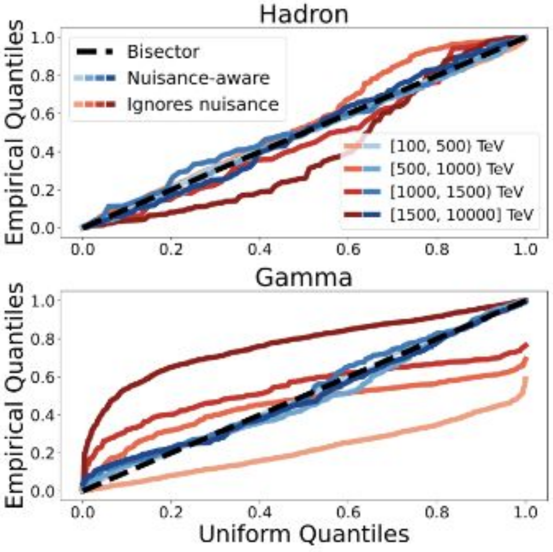
Song, et al
Nat Biotechnol (2023)

Application: Single-Cell RNA Sequencing

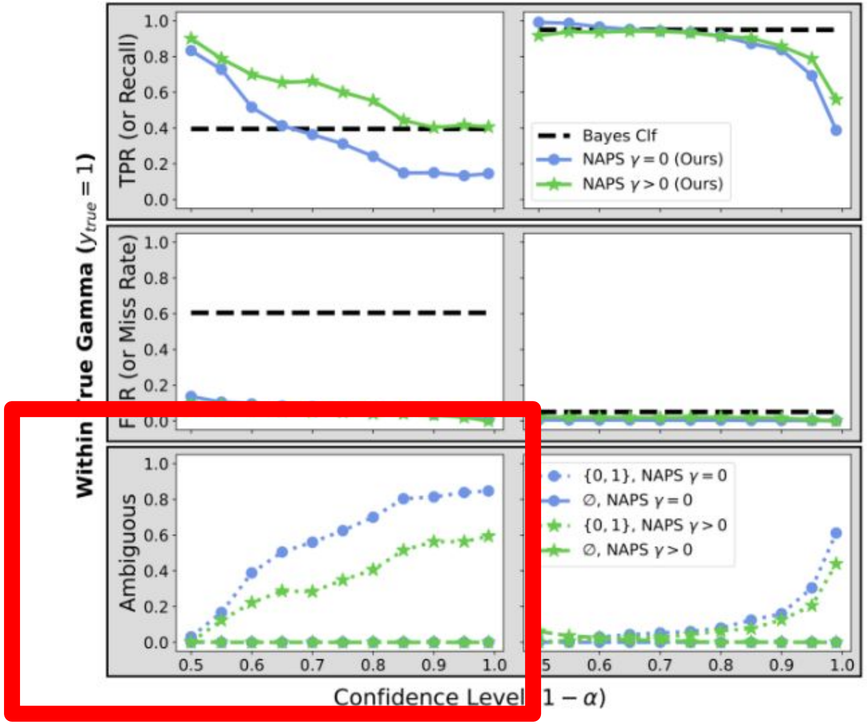


Application: Atmospheric Cosmic-Ray Showers

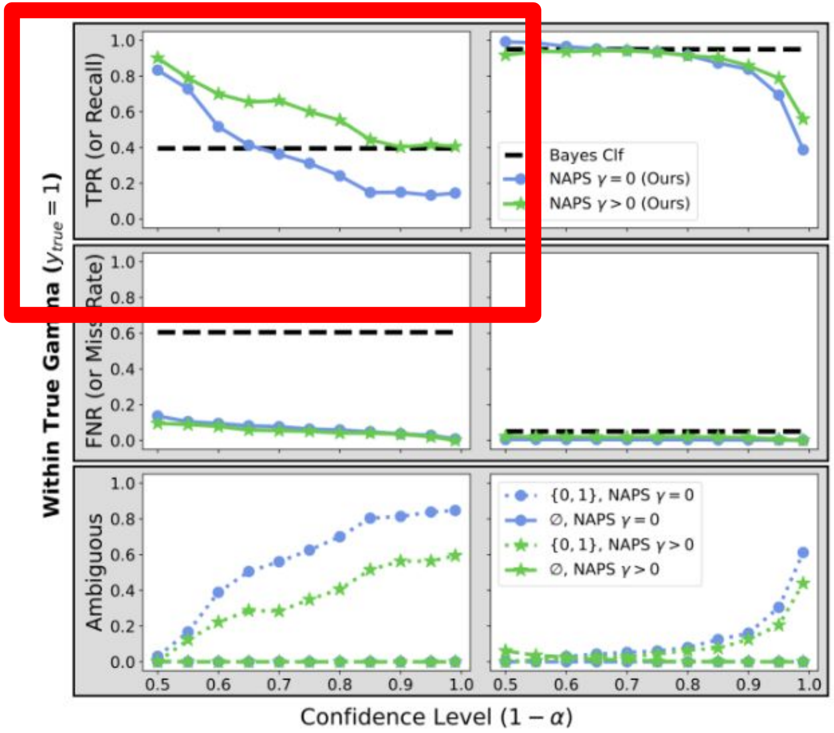




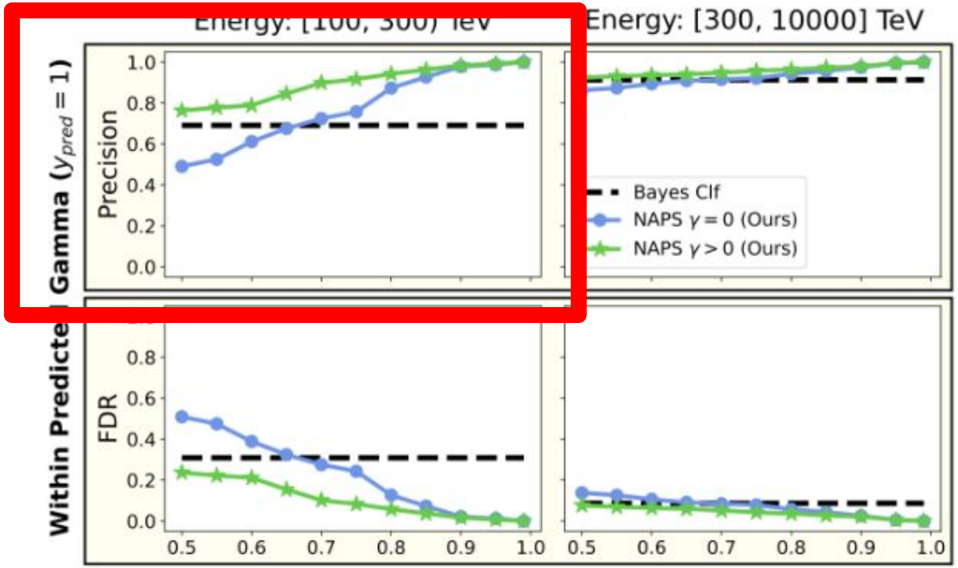
Application: Atmospheric Cosmic-Ray Showers



Application: Atmospheric Cosmic-Ray Showers



Application: Atmospheric Cosmic-Ray Showers



Final Remarks

- GLS models how (X,Y) changes
- Applications to SBI/LFI
- Can be used for classification if training data with (X,Y,$\boldsymbol{\nu}$) is available

arXiv > stat > arXiv:2402.05330

Search...

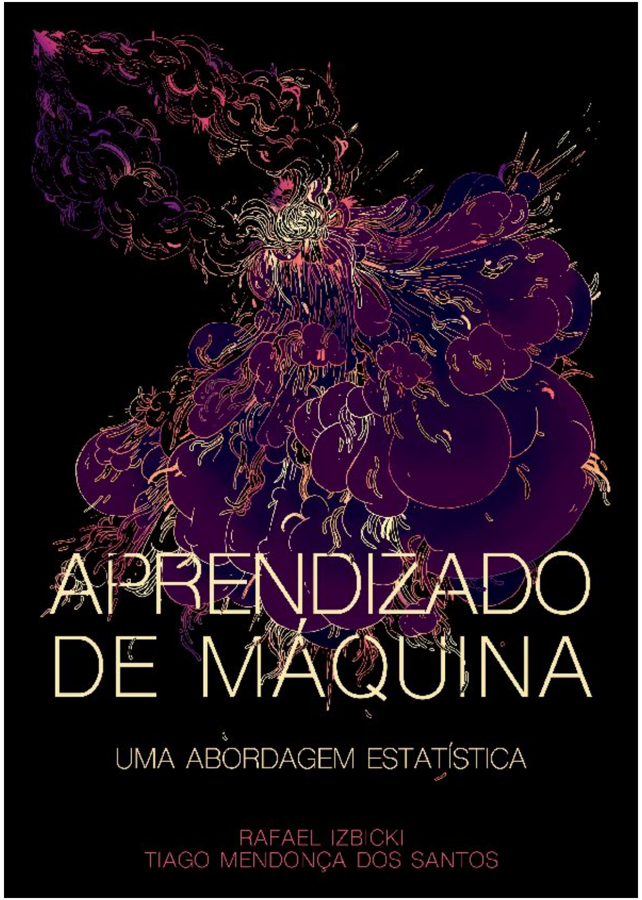
Help | Adva

Statistics > Machine Learning

[Submitted on 8 Feb 2024]

Classification under Nuisance Parameters and Generalized Label Shift in Likelihood-Free Inference

Luca Masserano, Alex Shen, Michele Doro, Tommaso Dorigo, Rafael Izbicki, Ann B. Lee



www.rafaelizbicki.com/ame

PIPGEs

Programa Interinstitucional
de Pós-Graduação em
Estatística UFSCar - USP

www.pipges.ufscar.br

Thanks!

rafaelizbicki@gmail.com

Statistical Machine Learning Lab

www.small.ufscar.br