ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

UIUC updates on SVA1 Gold

TPZ photo-z PDFs, probabilistic S/G and $N(z)$

Matías Carrasco Kind
Edward Kim
Robert Brunner

Department of Astronomy
University of Illinois

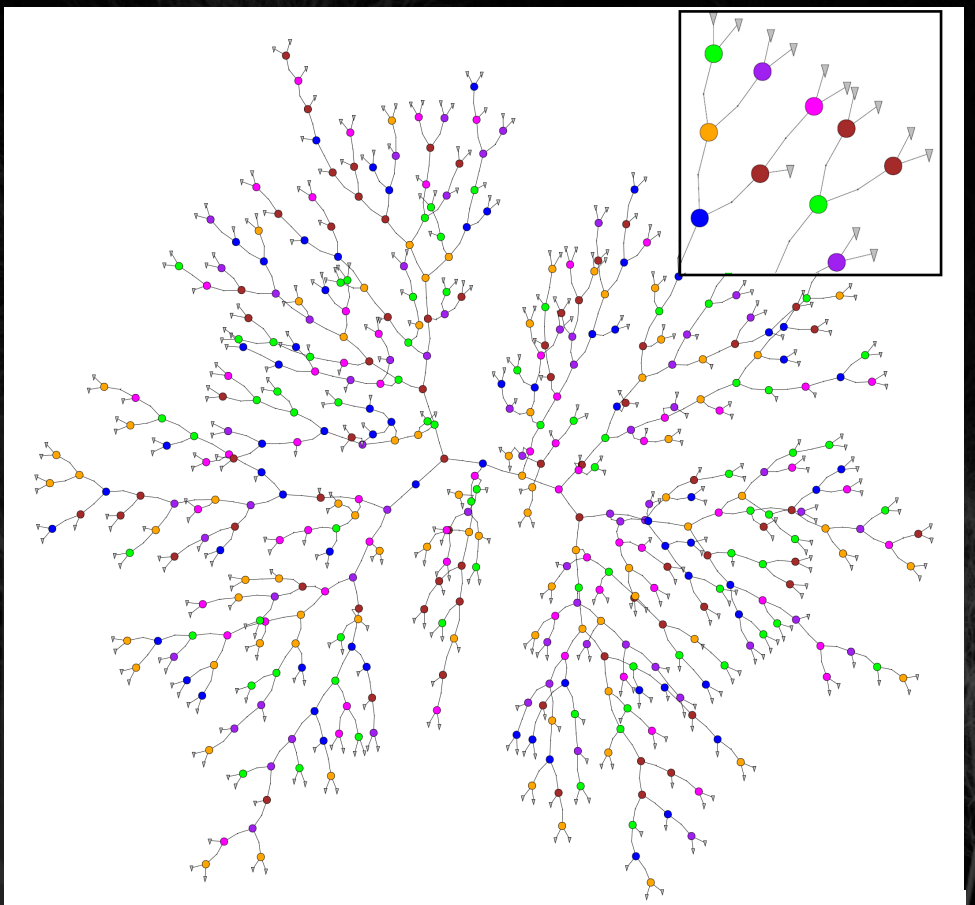
DES Telecon July 28^{th}, 2014

Outline

- TPZ for photo-z and S/G
- PDF sparse representation
- "Fast" $N(z)$ PDF reconstruction
- Usage (practical issues)
- To be done

Photo-z PDF estimation: TPZ

- TPZ (Trees for Photo-Z) is a supervised machine learning code
- Prediction trees and random forest
- Incorporate measurements errors and deals with missing values
- Ancillary information: expected errors, attribute ranking and others
- Application to the S/G

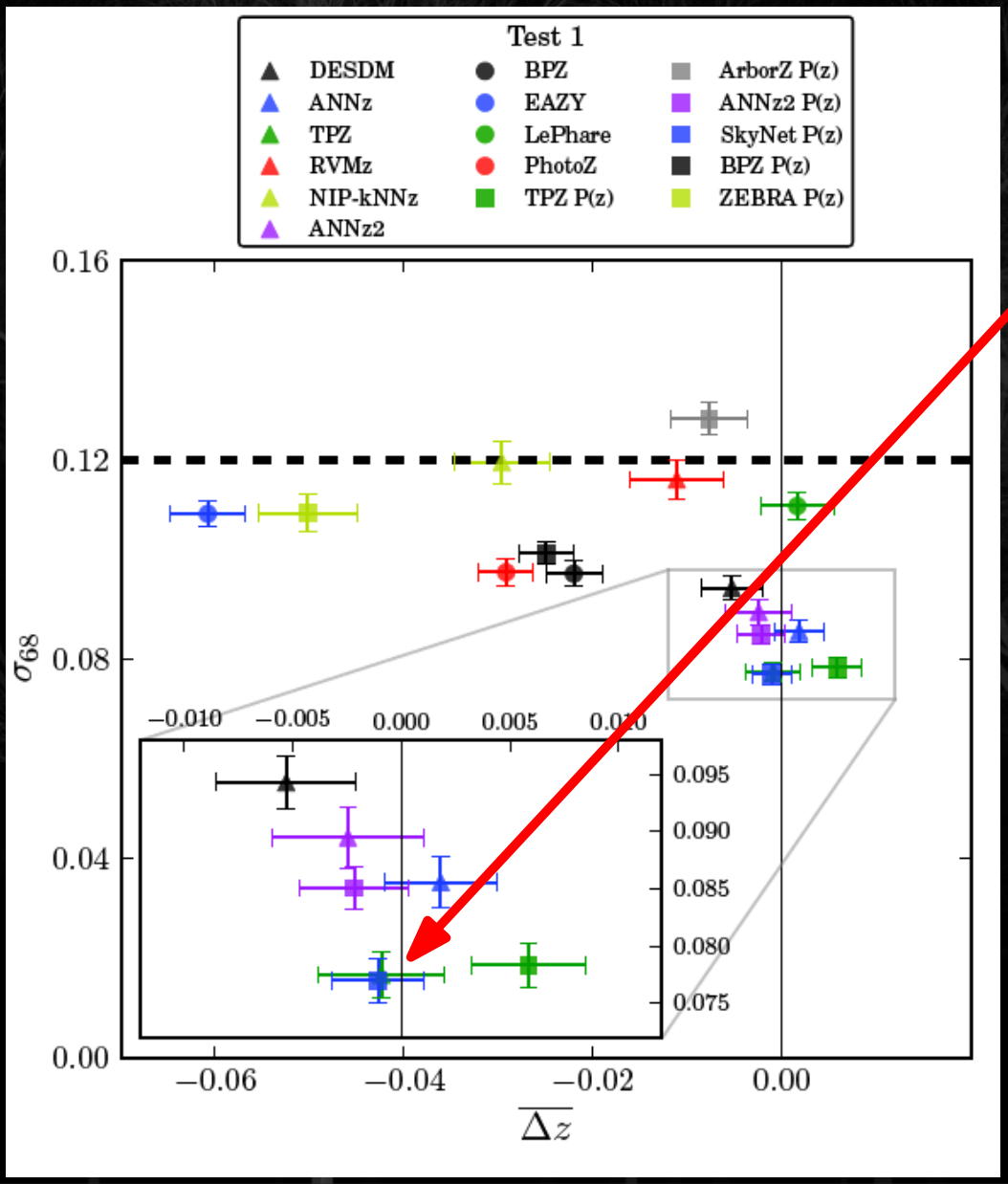


Carrasco Kind & Brunner 2013a (MNRAS, 432, 1483)

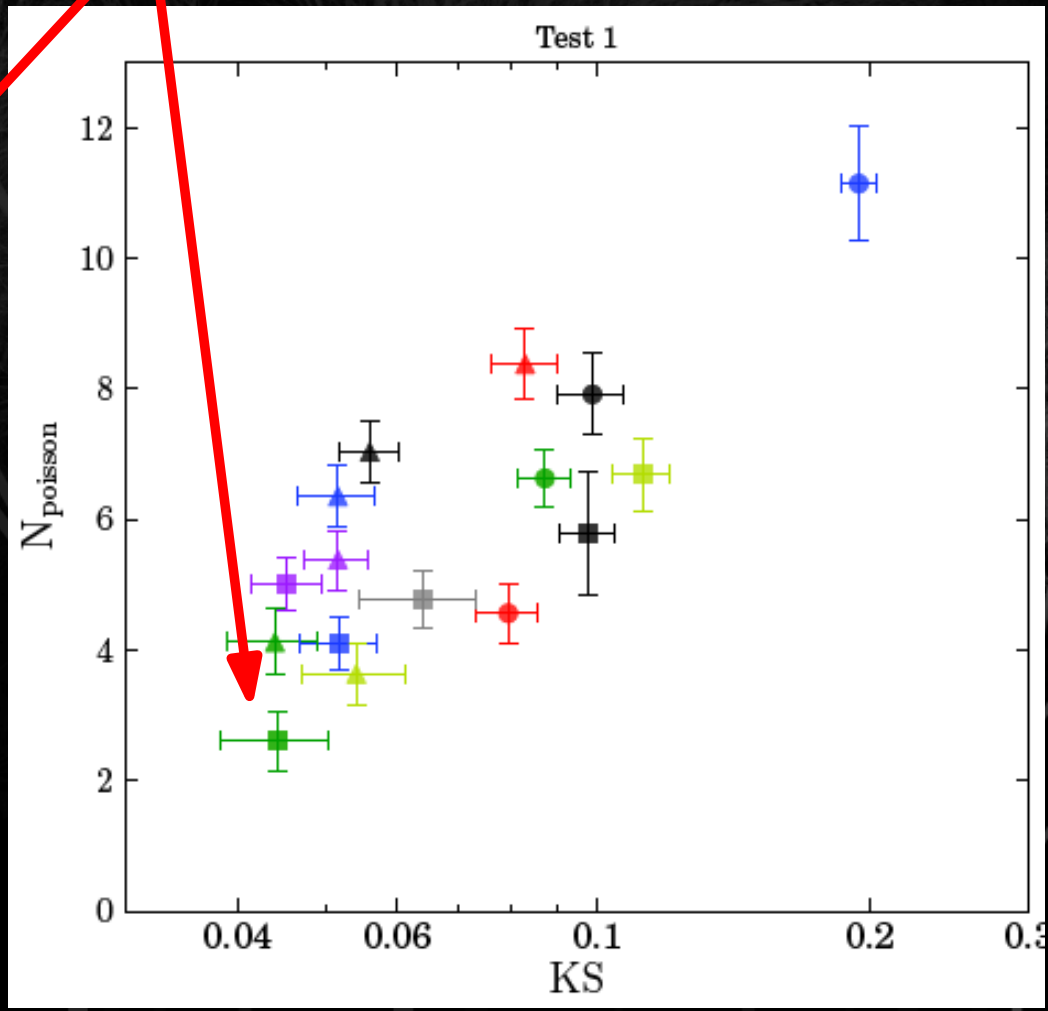
http://lcdm.astro.illinois.edu/code/mlz.html

TPZ photo-z on DES SV data

Photo-z DES paper $\rightarrow$ Sanchéz et al., 2014, arXiv:1406.4407



TPZ

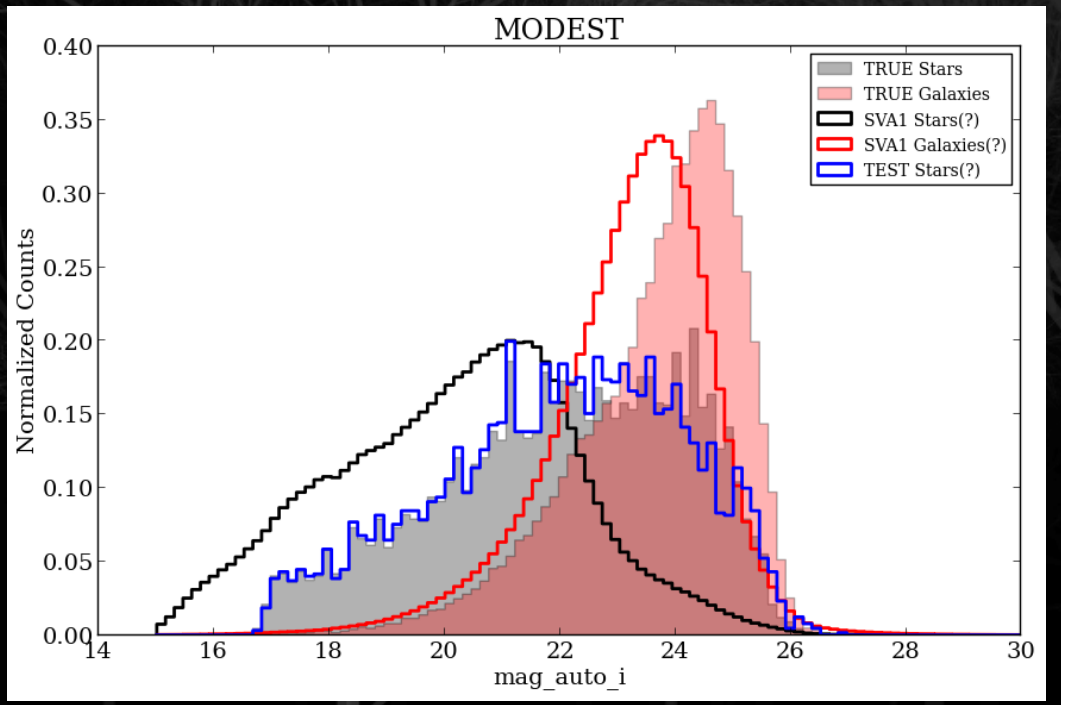


TPZ S/G on DES SVA gold data

S/G Challenge (Alex, Nacho et al.)

https://cdcvs.fnal.gov/redmine/projects/des-sci-verification/wiki/SVA1_Gold_v10_SG_Validation

TPZ seems more robust than other training based approaches and provides probabilistic classes



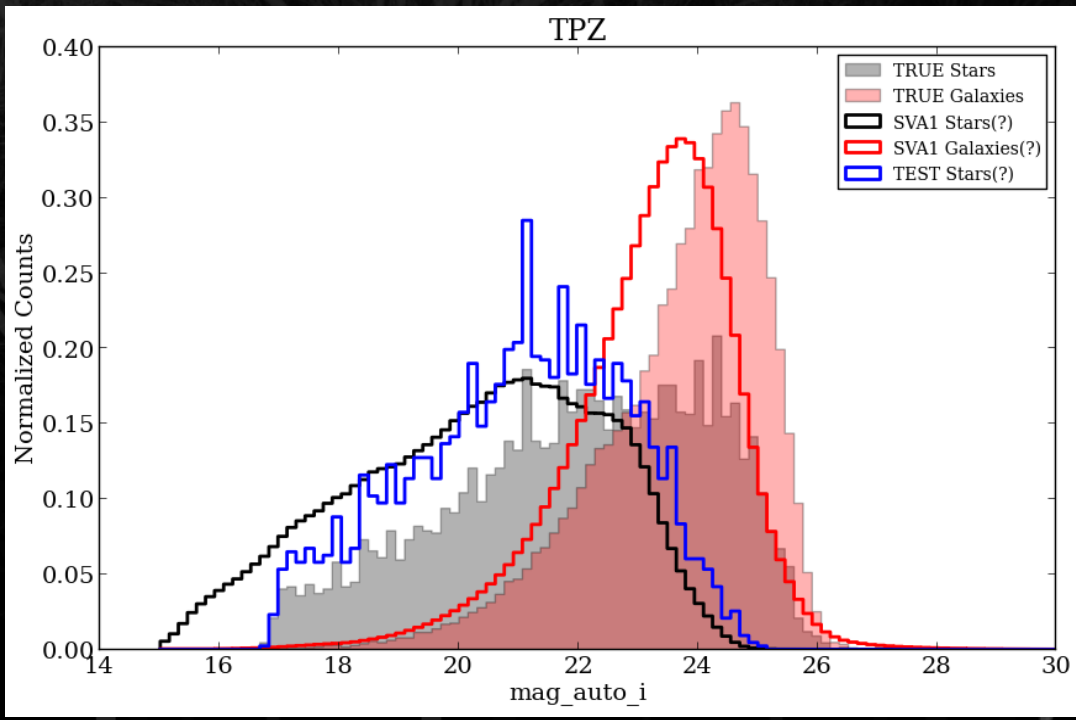


Photo-z PDF storage: Sparse representation

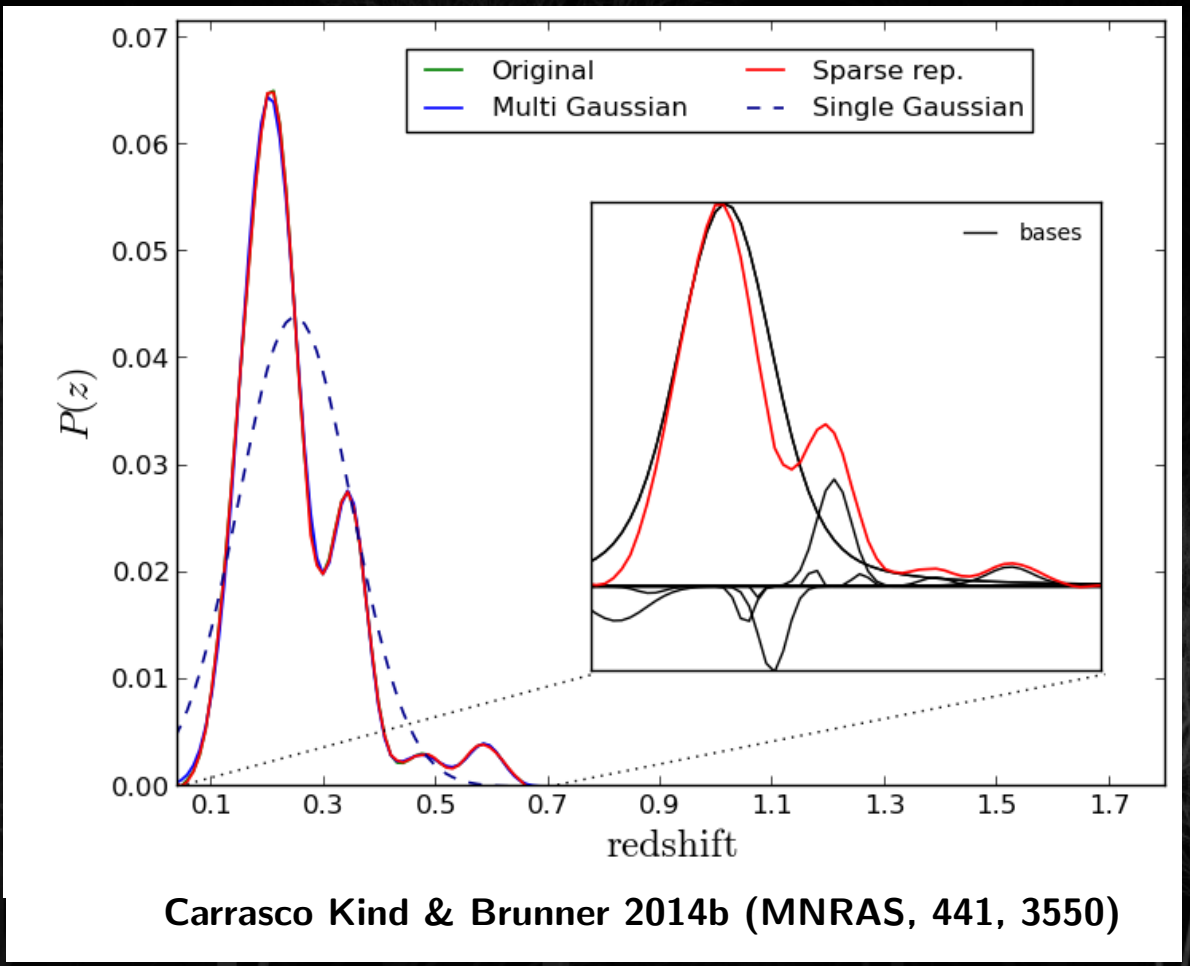
Use Gaussian and Voigt profiles as bases, need $N^2_{\rm original}$ bases

Find basis and amplitud to reduce residual on each step

With only 10-20 bases achieve 99.9 % accuracy

Use 32-bits integer per basis, compression

Store Multiple PDFs



Carrasco Kind & Brunner 2014b (MNRAS, 441, 3550)

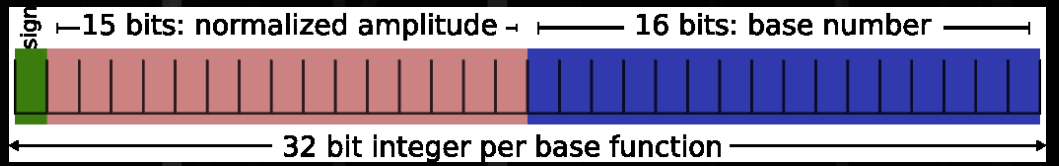
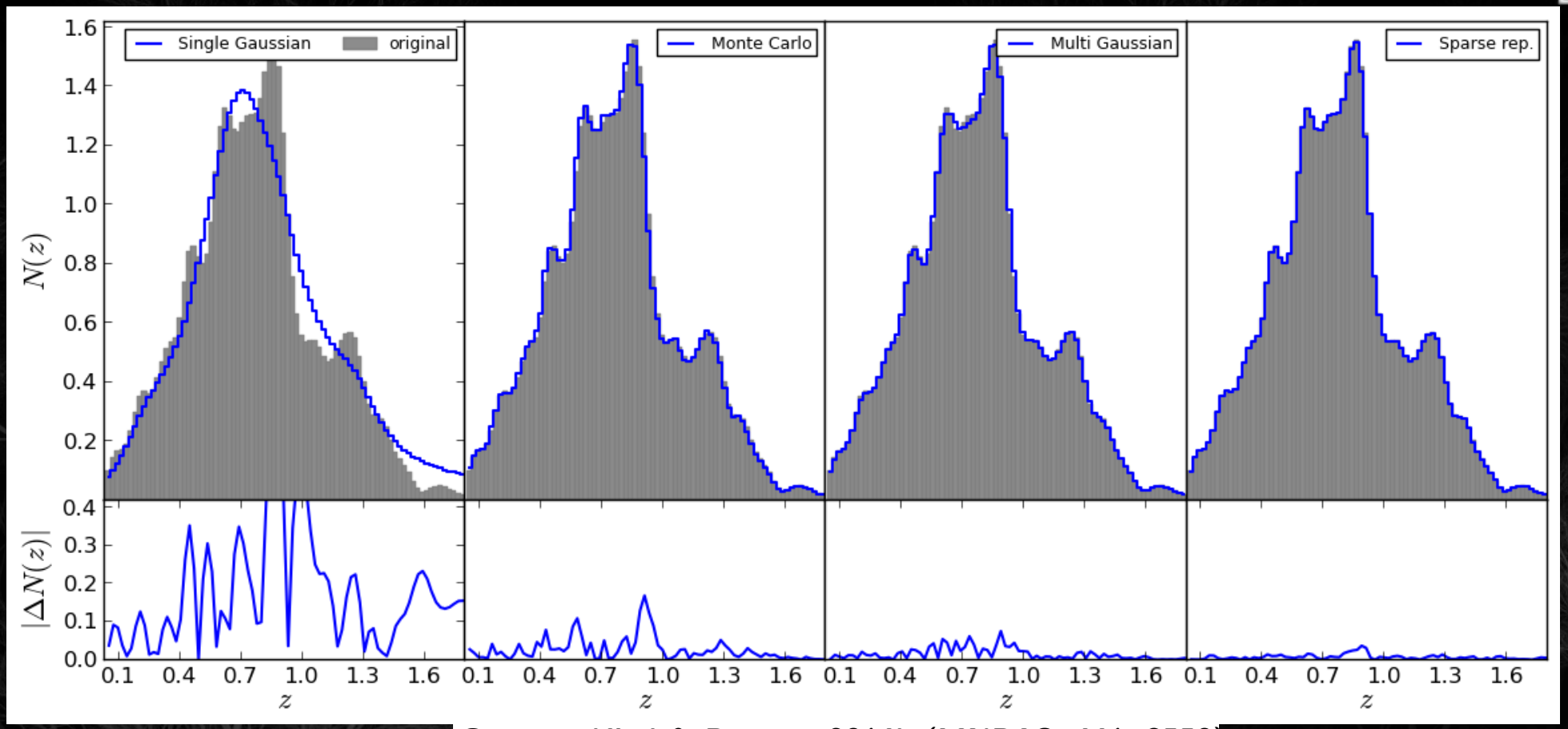


Photo-z PDF storage: Sparse representation



Carrasco Kind & Brunner 2014b (MNRAS, 441, 3550)

For PDFs with less than 4 peaks 5-10 points should be sufficient

Sparse representation gives more accurate and more compressed representation for N(z), 99.9% accuracy with 15 points (200 points originally)

Photo-z PDF application: $N(z)$

By definition:

$$N(z) = \sum_{k=1}^{N} \int_{z-\Delta z/2}^{z+\Delta z/2} P_k(z) dz$$

Using sparse representation, we represent each PDF pz_k as:

$\mathbf{pz}_k \approx \mathbf{D} \cdot \boldsymbol{\delta}_k$ $\mathbf{D}$ is the dictionary, $\boldsymbol{\delta}_k$ is the sparse vector, then

$$N(z) = \sum_{k=1}^{N} \boldsymbol{\delta}_k \cdot \int_{z-\Delta z/2}^{z+\Delta z/2} \mathbf{D} dz$$

Only bases are integrated

by precomputing:

$$\boldsymbol{\delta}_N = \sum_{k=1}^{N} \boldsymbol{\delta}_k \qquad \mathbf{I_D}(z) = \int_{z-\Delta z/2}^{z+\Delta z/2} \mathbf{d}_j dz \qquad j = 1, 2, \ldots, n$$

$N(z)$ is reduce to a simple dot product

$$\boxed{N(z) = \mathbf{I_D}(z) \cdot \boldsymbol{\delta}_N}$$

New compressed catalogs

2 new SVA1 catalogs, one for the sparse P(z) in 20 points instead of 200 (original) and one with basic information with following columns:

COADD_OBJECTS_ID
MAG_AUTO_I
MAG_DETMODEL_G
MAG_DETMODEL_R
MAG_DETMODEL_I
MAG_DETMODEL_Z
RA
DEC
MODEST_CLASS

From Eli's gold catalogs

Added from TPZ

TPZ_SG_CLASS (Probabilistic S/G, 0 for galaxies 1 for stars)
TPZ_ZPHOT (Mean of each photo-z PDF)
TPZ_ZCONF (Quality value similar to the ODDS parameter)

Creating a mask to select objects for $N(z)$

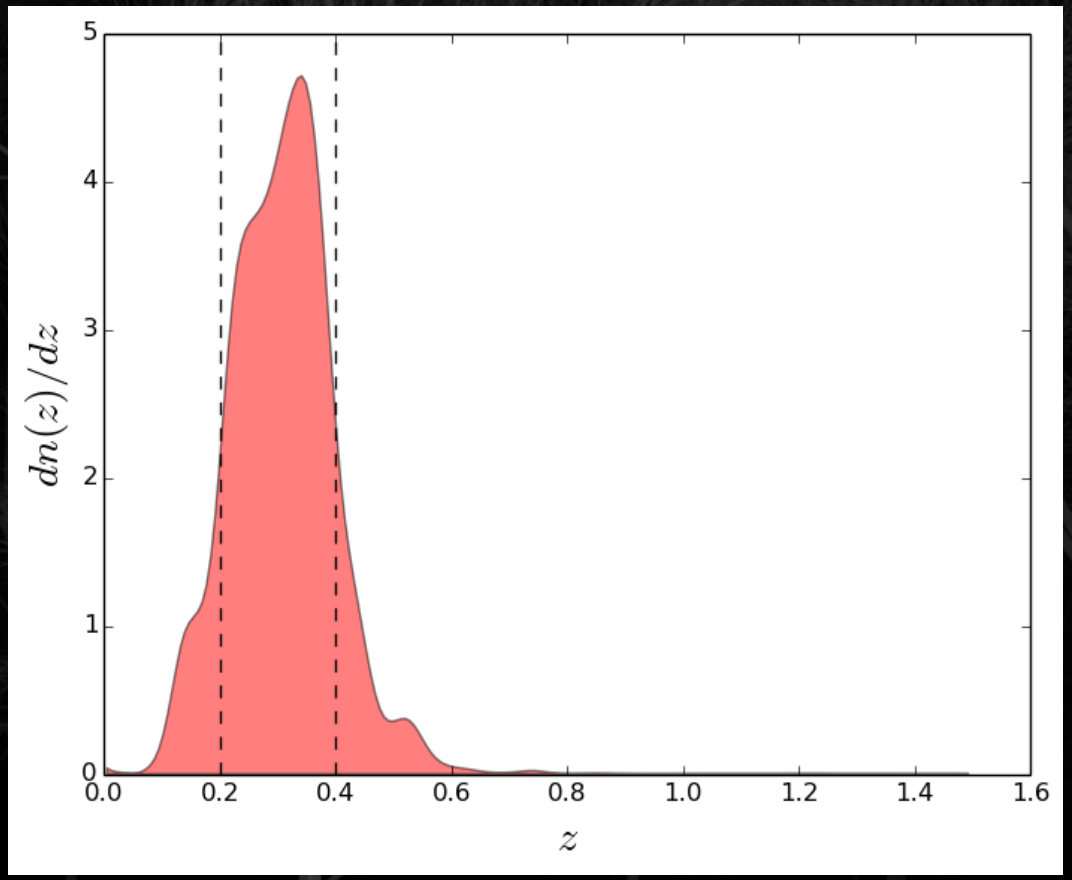
https://cdcvs.fnal.gov/redmine/projects/des-lss/wiki/LSS_related_bench-mark_galaxy_sample

Piece of the python code to be modified for your defined cuts, default values are LSS Benchmark

```
# This is the mask used to get N(z) and galaxies in general
# for sva1_gold_1.0 all the objects
# What's there is what's included
# For a range of values used a [min_val, max_val]
# For equalities used [value]
# You should use only one of the S/G keys AND COMMENT THE OTHER ONE
# For TPZ_SG_CLASS the range goes from 0 (galaxies) to 1 (stars)
#The default values are for the LSS benchmark cuts
Mk={}
Mk['MAG_AUTO_I']=[18,22.5]
Mk['RA']=[60,95]
Mk['DEC']=[-62,-40]
Mk['MODEST_CLASS']=[1] # USE MODEST_CLASS or TPZ_SG_CLASS between 0 (galaxies) and 1 (stars)
#Mk['TPZ_SG_CLASS']=[0.0,0.8] #TPZ PROB S/G Classification
Mk['TPZ_ZCONF']=[0.6,1.1] #ZCONF, goodness of the PDF, quality cut
Mk['TPZ_ZPHOT']=[0.2,0.4] #REDSHIFT BIN FROM TPZ
Mk['MAG_DETMODEL_G-MAG_DETMODEL_R']=[0,3]
Mk['MAG_DETMODEL_R-MAG_DETMODEL_I']=[0,2]
Mk['MAG_DETMODEL_I-MAG_DETMODEL_Z']=[0,3]
```

Example Usage

```
$ python compute_Nz.py
```



- Get python scripts and the 2 catalogs in the same folder
- Modify compute_Nz.py to desired mask and run it to generate N(z)
- Inside file there is example routines to retrive PDF and more info
- Soon more capabilities (like get PDFs and/or N(z) given ids

To be done.... (Improvements)

- Use last spectroscopic catalog matched with Gold to re-run TPZ
- Round 4 for S/G separation challenge, new data will provide better results
- Cross analysis of systematics using photo-z PDFs and probabilstic S/G

THANKS!

References

- TPZ paper (Carrasco Kind & Brunner) : MNRAS, 432,1483
- Sparse representation (Carrasco Kind & Brunner) : MNRAS, 441, 3550
- Photo-z DES paper (Sanchéz et al., 2014) arXiv:1406.4407
- Link to catalogs and code:

https://cdcvs.fnal.gov/redmine/projects/des-photoz/wiki/TPZ_photo-z_PDF_and_N(z)_using_sparse_representation_SVA1_Gold

https://github.com/mgckind/get_Nz

Questions?

Matias Carrasco Kind
University of Illinois
mcarras2@illinois.edu
https://sites.google.com/site/mgckind/