Problems and Solutions for LSST Shape Measurement

The Lighthouse People and The LSST Dark Energy Science Collaboration

ABSTRACT

LSST weak lensing science has unprecedented requirements for the modelling of the LSST point spread function and the accurate measurement of galaxy shapes in the face of blending. In this document, we describe the results of a workshop on these issues held at Point Montara in February 2017. We discuss available solutions for the PSF modelling and shape measurement, lessons learned from their use in the Dark Energy Survey, remaining open issues, and progress and plans towards fixing those. In addition, we lay out a strategy for handling multi-epoch image data in an API useful with present weak lensing image analysis codes and a framework for validating PSF and shape measurement through image simulations.

Subject headings: latex: templates, papers: awesome

1. Introduction

We skip the introduction for now, but keep an introduction to this LaTeX class for reference.

This is a paper and note template for the LSST DESC (Ivezic et al. 2008; LSST Science Collaboration 2009; LSST Dark Energy Science Collaboration 2012). You can delete all this tutorial text whenever you like.

You can easily switch between various LaTeX styles for internal notes and peer reviewed journals. Documents can be compiled using the provided `Makefile`. The command `make` with no arguments compiles `main.tex` using the `lsstdescnote.cls` style. If you want to upgrade your Note into a journal article, just choose a journal name, between `make apj` (ApJ preprint format), `make apjl` (which uses the `emulateapj` style), `make prd`, `make prl`, and `make mnras`.

There are a number of useful LaTeX commands predefined in `macros.tex`. Notice that the section labels are prefixed with `sec:` to allow the use of the `\secref` command to reference a section (i.e., Section 1). Figures can be referenced with the `\figref` command, which assumes that the figure label is prefixed with `fig:`. In Figure 1 we show an example figure. You'll notice that the actual figure file is found in the `figures` directory. However, because we have specified this directory in our `\graphicspath` we do not need to explicitly specify the path to the image.

The `macros.tex` package also contains some conventional scientific units like Å, GeV, $M_{\odot}$, etc. and some editorial tools for highlighting **issues**, **text to be checked**, *comments*, and new additions.

Similar to the figure before, here we have included a table of data from `tables/table.tex`. Notice that again we are able to reference Table 1 with the `\tabref` command using the `tab:` prefix. Also notice that we haven't needed to specify the full path to the table because in the `Makefile` we include `./tables` directory in the `$TEXINPUTS` environment variable.

Equations appear as follows, and can be referred to as, for example, Equation 1 – just as for tables, we use the `\eqnref` command using the `eqn:` prefix.

$$\langle f(k) \rangle = \frac{\sum_{t=0}^{N} f(t,k)}{N} \tag{1}$$

Figure 1 shows an example figure, referred to with the `\figref` command and the `fig:` prefix.

If you are planning on committing your paper to GitHub, it's a good idea to write your tex as one sentence per line. This allows for an easier `diff` of changes. It also makes sense to think of latex as *code*, and sentences as logical statements, occupying one line each. Each line must "compile" in the mind of the reader.

2. MEDS: Multi-epoch data structures

write why we need this: unified API for PSF modelling / shape measurement / photometry codes to access single frame image, weight, astrometry and PSF information

2.1. High-order instrumental astrometric distortions in MEDS

Gary, Troxel, Mike, Erin: describe how this is implemented

The MEDS python class would allow for flexibly swapping out the WCS in the input file FITS header by something more elaborate if we have the base class provide access functions for the `cutout_row/col` variables (in addition to the Jacobian) **(Erin)**. A derived class could then implement these differently, e.g. by evaluating Gary's WCS **(Gary)**. Both this and the way shape

Table 1: Example table.

Column 1	Column 2	Column 3	Column 4
	deg	kpc	deg
Obj1	(0,0)	10	0.1
...	...	...	...
ObjN	(0,0)	10	0.1

Fig. 1.— An example figure: the LSST DESC logo, copied from `.logos/desc-logo.png` into `figures/example.png`.

measurement codes find the matching PSFEx model files could be implemented by an external simple table that maps exposure and CCD IDs to auxiliary filenames.

2.2. A MEDS API for LSST

Jim, Erin, Joe, Josh, Daniel: describe how this is implemented; it seems like an API that generates an object's MEDS information on the fly could be feasible

3. PIFF: PSFs in the Full FOV

write an introduction of why we need this: astrometric distortions -¿ WCS, coherent patterns over full FOV, Zernickes, better interpolation schemes

3.1. Gaussian Process Interpolation

Josh, Gary, Mike, Niall, Pierre-Francois, Ami: describe

4. Shape measurement

quick intro of lessons learned from DES Y1

4.1. BFD on real data

Gary, Daniel, Joe, Katie, Ami: describe

The two components missing for this are

- a variant of `simpleImage` (see `momentcalc.py` in the BFD repository) that can take multiple postage stamps of the same galaxies with their respective WCS registrations (in the form of the position of a centroid estimated in WCS and transformed to the postage stamp pixel system) and Jacobians, PSF models, and an estimate of the overall centroid in WCS **(Katie)**
- a function that can get these inputs to the new variant of `simpleImage` from a MEDS file (using the python `meds` or a derived class) **(Daniel)**

5. An image simulation pipeline for PSF and shape measurement validation

Joe, Mike, Erin, Troxel, Gary, Daniel, Niall, Ami, Katie: describe

5.1. Goals

Goals: Simulation engines primarily for validation. Win some Gin from Catherine Heymans.

The subsections below describe stages of this process and the options or considerations for it.

5.2. (Statistical) Requirements

It's important to define cosmology-driven requirements for the simulation, as these will set the simulation volume requirements and thus the computing requirements etc.. Particularly important to do this if we e.g. need to find/apply for more computing resources.

I've brain-dumped a few thoughts on a couple of different (but certainly related) approaches we could take to this, (i) how many galaxies/galaxy pairs do we need to test our desired statistcs to a given precision? (ii) How many realisations of our survey do we need test our desired statistics to a given precision? I think the second approach is probably simpler because a simulation of our survey will automatically (of course limited by the simulation realism) have distributions of certain relevant survey properties (e.g. noise, PSF) that match the real data.

We want to demonstrate that our shear estimation pipelines can recover unbiased shear-shear and tangential shear signals to better than $X_i\%$, $Y_i\%$ for in n_z redshift bins, i, for $0 < z < z_{\rm max}$ for some range of scales $\theta_{\rm min} < \theta < \theta_{\rm max}$ (we could also phrase in terms of l and C_l. This requires:

$$\langle(1+m_i)(1+m_j)\rangle(\theta) + \langle c_i c_j\rangle(\theta) / \langle\gamma\gamma\rangle(\theta) < X\%, \tag{2}$$

$$\langle n_{\rm lens}(1+m_i)\rangle < Y\%. \tag{3}$$

Ignoring the scale dependence, assuming a constant shear, and that we calculate m as

$$\langle m_i\rangle = \left\langle \frac{\gamma_{\rm obs} - \gamma_{\rm true}}{\gamma_{\rm true}} \right\rangle, \tag{4}$$

where the (possibly weighed) averaging is over all galaxies in redshift bin i.

$$\sigma_m = \frac{\sigma_e}{\gamma_{\rm true}\sqrt{N_{\rm gal}}} \tag{5}$$

We want e.g. $\sigma_{m_i} < 0.5X_i\%$, so e.g. for $X_i = 1\%$, $\gamma_{\rm true} = 0.01$, $\sigma_e = 0.2$, we'd need to simulate 1.6×10^7 galaxies per redshift bin.

So what if we are worried about scale dependence (why would we be? blending, spatially varying observing conditions?). Well then we can write some requriements for the number of simulated pairs which populate a given angular bin etc....

An alternative way to think about the requriements is to first assume that we want to simulated $N_{\rm sim}$ copies of our survey ($N_{\rm sim}$ could, but probably wouldn't be less than 1). This approach has the advantage that (to the extent that the simulated survey properties are realistic), the simulated survey properties (e.g. noise, PSF etc.) will have the same distributions as the data. If we want to know the accuracy to which our pipeline recovers some statistic to better than a fraction f of the shape noise errors on that statistic we have in the real data, then we require $N_{\rm sim} = 1/f^2$ realisations of our data.

Discussion points:

- What should f be? Note that most of the statistics we use are cosmic variance limitied on large scales, so beating down the shape noise may not be so relevant. We will know the 'cosmic variance' (or rather we'll know the particular realisation of the cosmological signal) in the simulation.
- How can we reduce $N_{\rm sim}$? Boost the shear signal - the stronger the simulate signal, the better fractional accuracy can be achieved? How about ring-test type tricks to reduce shape noise? Ideas for this may be needed to make this computationally feasible.
- What kind of shear fields should we use? If we use something realistic (e.g. from ray-tracing), how do we test that we're recovering it correctly, given that a given method will only use a subset of the galaxies...

5.3. Required Ingredients

5.3.1. Input Catalogues

- Actual galaxy population is a no - has been shown to be problematic
- Start from N-body Simulation - positions, galaxy types
- Need a module for mapping from BCC. Drawn image. Colour and size, halo, environment. Recipe from BCC outputs to galaxy appearance. Simple for DES, complicated for LSST. Have SEDs.
- Morphology, Bulge + Disc!, COSMOS real galaxies? + Lanusse method. Compromises?
- Need to convert this to an image in a given band
- Need to save shears for possible use later.

- Needs to go to high enough redshift and depth for deep fields

5.3.2. Shear Field

- From saved N-body results (see above)
- Constant or distributions across image

5.3.3. Survey Details

- Real data pointings
- Exposure times - from real survey
- Noise levels - from real survey
- Sky background - from real survey
- Deep data - Same process but with more exposures

5.3.4. True Astrometry

- Flat WCS
- Real image WCS
- Gary's WCS
- FITS header WCS
- One of the above + some error distribution

5.3.5. True PSF

- Estimated real ones from Piff
- Additional complexity, adding variation on smaller scales than
- Fixed values
- Colour-dependence of PSF. Full implementation 100 times slower. Could make it a function of a single colour parameter, linearly interpolated. Effective PSF from real SED? Function of (g-i). Intra-band resolution of PSF. Do this at n wavelengths. Like using a chunky SED.

5.3.6. Star Catalogue

- Gaia?
- Randomly, function of latitude.
- SEDs - some in galsim already, need more?

5.3.7. Artifacts

- Tape bumps
- CTI
- Brighter-fatter
- Non-linearity
- Non-convolutional things
- Image artifacts

5.3.8. Masking

- Real
- Real + artifacts

5.3.9. Single-epoch Rendering

- GalSim

5.3.10. Coadding

- Run full pipeline - i.e. run swarp
- Shortcut option: draw coadd directly. What does this miss? What is required to test this?

5.3.11. Detection & Segmenting

- Needed for lists of detected objects and segmentation masks
- Run full pipeline - SExtractor on coadd
- Generate object list and seg map from truth catalogs

5.3.12. Estimated PSF

- Find starting stars - on coadd using SExtractor? Can we do this without spread-model?
- Run PIFF
- Use truth

5.3.13. Estimated astrometry

- Cannot re-do the actual astrometry process
- Use the original FITS ones
- Use truth + some error term

5.3.14. Photometric Calibration

- Could place a small error on the truth - scale objects by some factor.

5.3.15. MEDS

- Make a MEDS file and run on it!

6. `Metacalibration` Response for Stars

Testing to see if we can cause positive responses for stars in a simulation. In some scenarios we see $< R >\sim 0.25$ in real data.

6.1. Variations in PSF

Consider the case where the PSF model used is accurate in the mean but for an individual galaxy the truth varies significantly. The response for stars at high S/N(> 100), and using forward modeling, is shown in figure 2. The same for adaptive moments, without PSF correction. The response has mean about 0.1 for forward modeling, but nearly zero for adaptive moments. In both cases the distribution is nearly gaussian, whereas in real data it tends to be highly asymmetric, with mode at $R \sim 0.5$.

Acknowledgments

We acknowledge financial and organizational support for our workshop from LSSTC and KIPAC and the hospitality of Hostelling International.

This is the text imported from `acknowledgments.tex`, and will be replaced by some standard LSST DESC boilerplate at some point.

REFERENCES

Ivezic, Z., Tyson, J. A., et al. 2008, ArXiv e-prints, arXiv:0805.2366

LSST Dark Energy Science Collaboration. 2012, ArXiv e-prints, arXiv:1211.0310

LSST Science Collaboration. 2009, ArXiv e-prints, arXiv:0912.0201

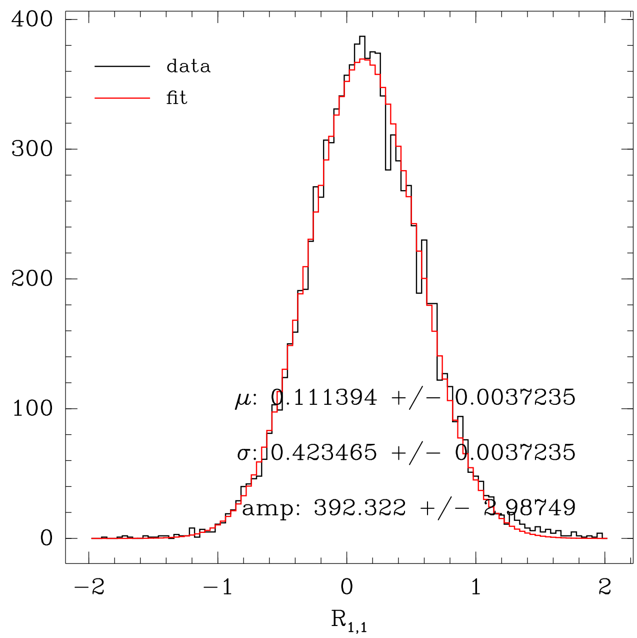

Fig. 2.— Response for high S/N stars when varying the PSF from object to object, but using the mean model when performing `metacalibration` operatioins, using the forward modeling estimator. Parameters of the best fit gaussian are shown.

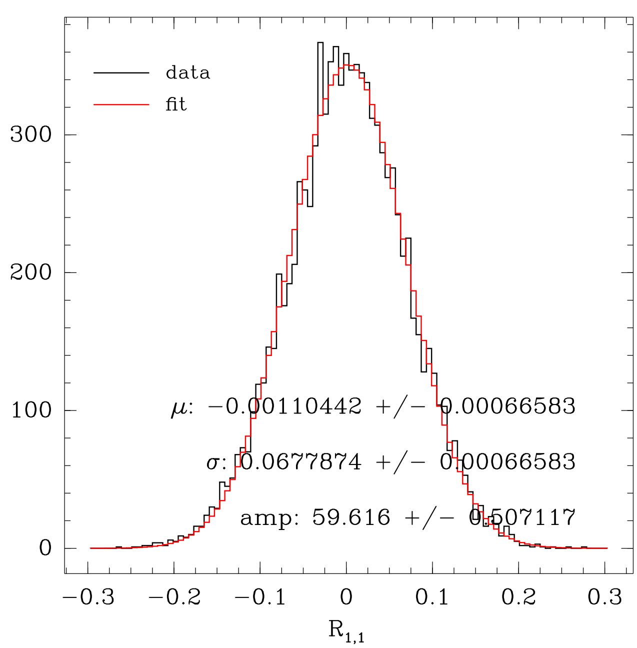

Fig. 3.— Same as figure 2, but adaptive moments estimator with no PSF correction