Title: LOR for the GPz PZ Estimator
Contributors: Peter Hatfield (Oxford)

Co-signers: Nathan Adams (Manchester), Ibrahim Almosallam (Saudi Information Technology Company), David Alonso (Oxford), Rebecca Bowler (Oxford), Matt Jarvis (Oxford), Stephen Roberts (Oxford), Natalia Stylianou (Leicester), Nijin Thykkathu (Oxford), Aprajita Verma (Oxford)

0. Summary Statement

This letter of recommendation describes the GPz photometric code (Almosallam 2016a, 2016b). The code has been benchmarked in data challenges for Rubin (Schmidt 2020 and the DESC Tomography Challenge) and Euclid (Desprez 2020), and used for science for a number of extragalactic data sets (Gomes 2018, Duncan 2018, Hatfield 2020). We are the group focussed in Oxford that has developed the code, and applied it to a range of extragalactic challenges

1. Scientific Utility

Broad utility in extragalactic science and cosmology – algorithm is not specific to a particular science goal. Usage to date has mainly been for producing catalogue redshifts for deep wide-field multi-wavelength surveys e.g. cross-matched LOFAR (Duncan 2018) and VIDEO (Hatfield 2020). These redshifts have then been used for a range of cosmological and extragalactic projects. Science projects of interest to our group for which we require photo-z include luminosity functions, galaxy clustering/HOD modelling, finding and modelling strong gravitational lenses, finding high-z (z>6) sources and cross-matching optical and radio surveys. GPz has been used for both for data sets with just optical bands, but also optical observations cross-matched with NIR data.

2. Outputs

Core algorithm, outputs a point estimate (μ), and a variance (σ^2). The variance has three components ($\sigma^2=\beta^{-1}_{*}+ \gamma + v$), which are also output if desired, uncertainty due to output noise (β^{-1}_{*}, e.g. galaxies with different redshifts but the same magnitudes), uncertainty due to input noise (γ, e.g. uncertainty due to error bars on the photometry), and uncertainty due to lack of training data in that part of parameter space (v).

More complex outputs beyond core outputs are possible e.g. a flags for high $v/(\beta^{-1}_{*}+ \gamma)$ values (indicating when algorithm is extrapolating).

3. Performance

Overall performance as with other machine learning codes will depend on spectroscopic training set and photometric bands available, but see data challenges for Rubin (Schmidt 2020) and Euclid (Desprez 2020) for comparison.

Compared to other machine learning codes GPz is well-rounded, typically performing well across a range of metrics, normally performing better on bias-like metrics, and metrics that assess the realism of uncertainty estimates. It is also typically a very fast running (both for training and prediction) algorithm, and is very easy to use. There is essentially only one parameter (m, the number of basis functions) that needs choosing for the training (compared with say the need to choose a network architecture for a neural network), which gives a trade-off between accuracy and prediction time (high m is more accurate but a longer run time, and vice versa), but empirical tests have shown that even this parameter doesn't require much fine tuning, and m~50 has been sufficient for almost all purposes.

GPz performance is perhaps weakest for galaxies that should have heavily multi-modal output pdfs (often due to degeneracy between the Lyman and 4000A break), as it outputs uni-modal Gaussian pdfs.

4. Technical Aspects

Scalability – **Will Meet**

Precise prediction time depends on exact implementation, number of bands, complexity permitted etc, but is typically 0.01-1 milliseconds per object on a laptop.

Inputs – **Will Meet**

Magnitudes/fluxes (and optionally the corresponding uncertainties). Does not require pixel level data. Galaxy shape parameter(s) (or any other properties) can be used (tested in Gomes 2018). Can also accept missing bands (e.g. if some of the galaxies are missing photometry for some of the passbands). Many codes impute the missing values by taking the average or similar, which is not quite accurate as the expectation of the function is not necessarily the function of the expectation. GPz handles this by marginalizing the missing values and computes the expected value in the absence of the unavailable information.

Outputs – **Will Meet**

Predictions are Gaussians (possibly truncated at z=0), so mode, mean, standard deviation, skewness, kurtosis, and 1%, 5%, 25%, 50%, 75%, and 99% limits can easily be calculated but do not provide any further information.

The uncertainty breakdown into three components as described in Section 2 are not currently envisaged as outputs in LSE-163, so would either have to be added as new outputs if considered useful, converted to a flag (e.g. some threshold of $v/(\beta^{-1}_{*}+ \gamma)$ values), or not used.

Storage Constraints – **Will Meet**

GPz output pdfs are Gaussian so thus completely determined by two numbers, so no storage of large pdf tables required. No spectral templates or calibration data required, just the training data, which will be fairly minimal and similar to what is required by other machine learning codes e.g. a catalogue of photometry (6 bands, potentially with 6 corresponding uncertainties=12 columns, with possibly more from matched sources or size data) and matched spectroscopic redshifts (1 column) for as many sources are available (likely in the region of ~10^5).

The size of the trained model depends on size of training set and number of bands, but is typically represented by ~10,000 parameters.

External Data Sets – **Will Meet**

GPz will require a spectroscopic training set (e.g. photometry with cross-matched spectroscopic redshifts) as per other machine learning based photometric redshift codes.

Estimator Training and Iterative Development – **Will Meet**

Algorithm is very robust and should not require extensive development. However Oxford can input to the Data Management team during preliminary processing and development.

Computational Processing Constraints – **Will Meet**

Will only operate on measurements in the LSST Object catalogue. GPz has a small computational footprint, and has already been successfully folded into the Euclid pipeline.

Implementation Language – **Will Meet**

Pure python, C++ and Matlab versions currently exist. A python wrapper around the C++ version has almost finished development. This version will be incorporated into RAIL as part of PZ WG Deliverables in the LSST-DESC Science Roadmap.

Maintenance – **Will Meet**

Once fully tested and implemented, GPz should be require very little maintenance/updating. However Oxford will manage any ongoing development. Specialist knowledge required to use GPz is relatively little.