GSAS_USE

Anton Gagin and Igor Levin

This is an extension to the *GSAS-II* Rietveld package *GSAS_USE* (Bayesian Statistics Approach to Accounting for Unknown Systematic Errors), written and maintained by Anton Gagin (anton.gagin@nist.gov, av.gagin@gmail.com.)

GSAS_USE addresses the effects of systematic errors in Rietveld refinements. The errors are categorized into multiplicative, additive, and peak-shape types. Corrections for these errors are incorporated into using a Bayesian statistics approach, with the corrections themselves treated as nuisance parameters and marginalized out of the analysis. Structural parameters refined using the proposed method represent probability-weighted averages over all possible error corrections. See Gagin, A. & Levin, I. (2015). *Accounting for Unknown Systematic Errors in Rietveld Refinements: A Bayesian Statistics Approach. J. Appl. Cryst.* **xx**, xxx-xxx (in progress) for details.

The current version has been tested with *GSAS-II* version 0.2.0, revision 1852.
For details of the *GSAS-II* package, see Toby, B. H. & Von Dreele, R. B. (2013). *J. Appl. Cryst.* **46**, 544-549, or visit their website.

Table of Contents

Installation

To apply this patch, place the ***patchSystErrors*** folder in your *GSAS-II* local folder and run **__apply_patch__.py**, or print

```
execfile('__apply_patch__.py')
```

in a python command line interpreter. If everything correctly, the following message will be displayed

```
### based on Diff, Match and Patch Library
###          http://code.google.com/p/google-diff-match-patch/
###          by Neil Fraser
###          Copyright 2006 Google Inc.



--------------------------
This script will patch your current version of the GSAS-II package

Begin [y/n]?
```

Type `y` and follow the instructions.

Folder ***originalOld*** contains some of the original *GSAS-II* source files under revision 1852. Folder ***modifiedOld*** contains our modification of these files. The script copies the source files from your current revision of *GSAS-II* into the ***originalNew*** folder. Before applying the patch please ensure that the local folder with *GSAS-II* contains the original *GSAS-II*-files and not the modified versions! **___apply_patch___.py** calculates the patch from the difference between the files in the ***originalOld*** and ***modifiedOld*** folders, applies this patch to the files in the ***originalNew*** folder, and writes the results to the ***modifiedNew*** folder (as well as to the *GSAS-II* local folder.) To restore the original *GSAS-II*-files, run **___restore_original___.py**.

Usage

After the patch has been applied, start *GSAS-II* normally. In **Controls** menu specify the correction parameters. If several histograms are refined simultaneously, list these parameters, separated by commas, in the order corresponding to the order of the histograms. If you wish to the same value of the parameter for all the histograms, enter a single number. Set *E_mu*, *E_beta* or s to zero, if you do not want to apply a particular correction (multiplicative, additive, or peak-shape.)

If you select *Estimate optimal k_mu?*, the *Prior factor k_mu* field will be set to `optimal`. The same is true for the *Estimate optimal k_beta?* and *Prior factor k_beta* fields. Deselecting *Estimate optimal k?* will restore the previous value in *Prior factor k*.

If you click on *Correlation length l_delta* field, the *estimate it as FWHM /* field will be set to `none`, and vice versa. The same is true for the fields *Stdev sigma_delta* and *estimate it as l_delta/*.

To start a Bayesian-corrected refinement, select **Calculate/Refine** in the *GSAS-II* data tree window.

Description

- The multiplicative correction $\mu(x)$ is approximated by a set of *E_mu* cubic spline functions $\phi_j^{(\mu)}(x)$

  $$\mu(x) = \sum_{j=1}^{E_\mu} \left(1 + c_j^{(\mu)}\right) \phi_j^{(\mu)}(x),$$

  where $c_j^{(\mu)}$ are the spline coefficients. Spline-knot positions are selected equidistantly.
  The scaling parameter *k_mu* reflects the strength of the restriction on closeness of the multiplicative correction to unity. It can be estimated by the program from the residual of a standard fit (no corrections), if *Estimate optimal k_mu?* is selected.

- The additive correction is approximated using a set of *E_beta* cubic spline functions $\phi_j^{(\beta)}(x)$

  $$\beta(x) = \sum_{j=1}^{E_\beta} c_j^{(\beta)} \phi_j^{(\beta)}(x).$$

  The scaling parameter *k_beta* reflects the strength of the smoothness restriction on the additive correction.

- A diffraction profile is corrected by varying x-coordinates of the individual points of a diffraction curve. A probability of each 'move' δx is calculated as

  $$p(\delta x) \propto \exp\left(-\frac{1}{2}\delta x^T \Sigma_\delta^{-1} \delta x\right),$$

  where the covariance matrix Σ_δ^{-1} is defined as

  $$\Sigma_{ij}^{(\delta)} = \sigma_\delta^2 \exp\left(-\frac{1}{2}\left(\frac{x_i - x_j}{l_\delta}\right)^2\right).$$

The scaling parameters *sigma_delta* and *l_delta* describe the standard deviation for the correction and correlation length for the point coordinates, respectively. *l_delta* can be estimated from the characteristic FWHM values for the diffraction peaks (which depend on x) as $FWHM/p1$, where $p1$ can be any real number. For a multi-phase refinement, if estimated from the FWHM, *l_delta* is calculated as a an average weighted by the number of peaks for all the phases.
sigma_delta can be estimated from the *l_delta* value(s) as $l\_delta/p2$, where $p2$ can be any real number.
To reduce the computational complexity (e.g. one may get an out-of-memory error for extremely large histograms) and speed the calculations, the fitted x-range is divided into s independent segments.

- The iterative procedure works as follows:
  - a standard fit is performed
  - a Bayesian-corrected fit is performed
  - the optimal corrections are calculated and applied to the experimental data
  - a Bayesian-corrected fit is repeated

The second Bayesian-corrected fit is prone to overfitting becouse it uses the same correction parameters as have been already applied to the data. Therefore, we advise to limit the use of the iterative option to cases of large systematic errors.

Example

- Download the example files for a 'Combined X-ray/CW-neutron refinement of PbSO4' from the *GSAS-II* tutorial. Perform the refinements as desscribed in the tutorial.
- Deselect all the refinable parameters except for the structural variables which include 3 lattice parameters, 11 sets of atomic coordinates, and 5 isotropic atomic displacement parameters. MAKE SURE to deselect **Background** and **Histogram scale factor**!
- For this example we want to correct all three types of errors. Set the *Number of knots E_mu* to

  ```
  15, 20
  ```

  (more splines are selected for the XRD data because it shows the worse residual). These number of knots can be increased up to

  ```
  30, 45
  ```

  but this will take longer to calculate. Set *Priot factor k_mu* to

  ```
  1, 1
  ```

- Set *Number of knots E_beta* to

  ```
  15, 20
  ```

  and select *Estimate optimal k_beta?*
- Set *Number of blocks s* to

  ```
  8, 8
  ```

  To estimate correlation lengths *l_delta* and standard deviations *sigma_delta*, type

```
1.5
```

in the *estimate it as FWHM /* and

```
2.0
```

in the *estimate it as l_delta /* fields, respectively.

- Select **Calculate/Refine** in the *GSAS-II* data tree window. The program will perform a standard least-squares fit followed by a Bayesian-corrected fit. The results will be saved in the **projectName.lst** file. The details of the Bayesian fit will be stored in the **projectName_cor_iHist.txt** files, where **iHist** is the histogram number.

What the patch does

- constant-wavelength and time-of-flight data
- single- and multi-phase systems
- automatic selection of the optimal values for the prior parameters *k_mu* and *k_beta*
- use of $FWHM(x)$ dependence to estimate *l_delta*
- Bayesian-corrected fits
- calculations of the most plausible (optimal) corrections
- plots relevant residuals and histograms