SICK: THE SPECTROSCOPIC INFEERENCE CRANK

ANDREW R. CASEY
Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge, CB3 0HA, UK; arc@ast.cam.ac.uk
Received 2014 August 25; accepted 2016 January 21; published 2016 March 23

ABSTRACT

There exists an inordinate amount of spectral data in both public and private astronomical archives that remain severely under-utilized. The lack of reliable open-source tools for analyzing large volumes of spectra contributes to this situation, which is poised to worsen as large surveys successively release orders of magnitude more spectra. In this article I introduce sick, the spectroscopic inference crank, a flexible and fast Bayesian tool for inferring astrophysical parameters from spectra. sick is agnostic to the wavelength coverage, resolving power, or general data format, allowing any user to easily construct a generative model for their data, regardless of its source. sick can be used to provide a nearest-neighbor estimate of model parameters, a numerically optimized point estimate, or full Markov Chain Monte Carlo sampling of the posterior probability distributions. This generality empowers any astronomer to capitalize on the plethora of published synthetic and observed spectra, and make precise inferences for a host of astrophysical (and nuisance) quantities. Model intensities can be reliably approximated from existing grids of synthetic or observed spectra using linear multi-dimensional interpolation, or a Cannon-based model. Additional phenomena that transform the data (e.g., redshift, rotational broadening, continuum, spectral resolution) are incorporated as free parameters and can be marginalized away. Outlier pixels (e.g., cosmic rays or poorly modeled regimes) can be treated with a Gaussian mixture model, and a noise model is included to account for systematically underestimated variance. Combining these phenomena into a scalar-justified, quantitative model permits precise inferences with credible uncertainties on noisy data. I describe the common model features, the implementation details, and the default behavior, which is balanced to be suitable for most astronomical applications. Using a forward model on low-resolution, high signal-to-noise ratio spectra of M67 stars reveals atomic diffusion processes on the order of 0.05 dex, previously only measurable with differential analysis techniques in high-resolution spectra. sick is easy to use, well-tested, and freely available online through GitHub under the MIT license.

Key words: methods: data analysis – methods: statistical – stars: fundamental parameters – Sun: fundamental parameters – techniques: spectroscopic

1. INTRODUCTION

Most of our understanding of astrophysics has been interpreted from spectra. Given how informative spectroscopic data is to our understanding of astrophysics, it is not surprising that there has been a substantial increase of publicly accessible spectra in the last decade. Large-scale surveys have driven this trend, each releasing in excess of hundreds of thousands of spectra in the last decade. Large-scale surveys have driven this data explosion, and it is not surprising that there has been a substantial increase of publicly accessible spectra in the last decade. Large-scale surveys have driven this trend, each releasing in excess of hundreds of thousands of spectra in the last decade. Large-scale surveys have driven this trend, each releasing in excess of hundreds of thousands of spectra in the last decade. Large-scale surveys have driven this trend, each releasing in excess of hundreds of thousands of spectra in the last decade. Large-scale surveys have driven this trend, each releasing in excess of hundreds of thousands of spectra in the last decade.

These spectra are acquired from different astrophysical sources to meet specific scientific objectives. They vary in wavelength coverage, resolution, and noise distributions. For these reasons many collaborations expend significant resources to produce bespoke analysis software for their science program. This usually impedes reproducibility, as many codes still remain closed-source nearly a decade after the original article was published (e.g., Lee et al. 2008), even when the data and results are publicly accessible. As a consequence any comprehensive literature comparison becomes impossible, as systematics are difficult to properly characterize without in-depth knowledge of the methods or access to the software.

In general there are three types of methods employed for spectral analysis: measuring the strengths of spectral features, pure data-generating models or template-matching methods. Approaches that measure spectral features (e.g., equivalent widths; Stetson & Pancino 2008) are inexpensive, but regularly encounter problems with blended (often hidden) lines or continuum placement (e.g., see Smiljanic et al. 2014, for a comparison of techniques). In these instances some subjective interaction, tuning, or ad hoc “calibration” is often invoked (Sousa et al. 2010; Kordopatis et al. 2013). Data-generating methods are computationally expensive, repeatedly producing model spectra during run-time (Valenti & Piskunov 1996). While accurate spectra are produced, the known covariances between stellar parameters are routinely ignored, leading to erroneous results (Torres et al. 2012). For template-matching methods (Recio-Blanco et al. 2006; Allende Prieto et al. 2008), synthetic spectra are generated once for a subset of permutations of astrophysical quantities, usually discretized across a grid. Although there are differences between these methods, the preparatory steps are usually the same: spectra are placed at rest-frame by calculating line of sight velocities, typically by cross-correlation, before being continuum-normalized to zero and one.

Credible uncertainties can be difficult to discern from these methods. This is because the uncertainties in the Doppler-shift, smoothing, sampling, and normalization steps are almost always ignored. These effects result in ill-characterized uncertainties in astrophysical parameters. In some cases the uncertainties are simply assumed to be approximately the same for all objects. This is an incorrect approach: there are few, if any, examples of homoscedastic datasets in astrophysics. The noise properties of each spectrum are different, and the parameter uncertainties (random and systematic) will differ for
every object. Consequently the uncertainties in astrophysical parameters by template-matching methods are generally found to either be incorrectly assumed, under-estimated, or at least ill-characterized.

In addition to affecting the uncertainties, the effects of redshift, continuum normalization, and smoothing will further bias the reported maximum-likelihood parameters. For example, experienced stellar spectroscopists will frequently differ in their decision of continuum placement even in the most straightforward cases (e.g., metal-poor stars). There are a number of controversial examples within the literature where a subjective (human) decision of continuum placement has significantly altered the scientific conclusions (e.g., see Kerzendorf et al. 2013, where this issue is discussed in great detail). The implications of these phenomena must be considered if we are to understand subtle astrophysical processes. Spectroscopy requires open-sourced objectivity. One should endeavour to incorporate these phenomena as free parameters into a generative model and infer them simultaneously with the astrophysical parameters.

In this article I present sick: a flexible, well-tested, MIT-licensed probabilistic software package for inferring astrophysical (and nuisance) quantities from spectroscopic data. sick employs approximations to data-generating models. Instead of modeling expensive astrophysical processes (e.g., stars, supernova, and any other interesting astrophysical processes) at run-time, sick approximates the model intensities from pre-computed grids/sets of spectra. Contributory phenomena (e.g., continuum, redshift, rotational broadening, spectral resolution) are included as free parameters within an objective scalar-justified model. This approach is suitable for a plethora of different astrophysical processes, allowing any user to easily construct a generative model using existing grids of published spectra, and infer astrophysical properties from their data. Aspects of the probabilistic model are described in Section 2. The analysis methodology is discussed in Section 3, and a toy model is presented in Section 4. In Section 5 I present a suitable scientific application that demonstrates the power of forward models with existing data, instead of existing subjective approaches. I conclude in Section 7 with references to the online documentation and applicability of the software.

2. THE GENERATIVE MODEL

The generative model described below is agnostic as to what the astrophysical parameters actually describe.1 Typical examples might be properties of supernova (e.g., explosion energies and luminosities), galaxy characteristics from integrated light, or mean plasma properties of a stellar photosphere. There are a plethora of spectral libraries (observed and synthetic) published for these types of applications (e.g., Le Borgne et al. 2004; Blondin & Tonry 2007; Palacios et al. 2010; Husser et al. 2013). All of these models are fully sick compatible.2

In an ideal world one would avoid spectrum approximations entirely and aim to solve the hydrodynamic and radiative transfer equations to produce accurate model spectra in real-time. This would certainly be a computationally expensive endeavour. It may also be unnecessary. In practice sick can be sub-classed (by inheriting from the BaseModel class) to produce more realistic spectra at run-time or to allow approximations within existing grids with different approaches. The code is designed to be flexible to suit a range of scientific objectives.

Let us assume that there exists a set of astrophysical parameters $\theta$ that I wish to infer from some data. I first must produce a spectrum of normalized model intensities (e.g., between 0 and 1) from an existing set (or grid) of spectra. The grid of spectra is expected to span a suitable range of $\theta$ values, but is not required to be regularly spaced in $\theta$. Indeed, high-quality observed spectra with irregularly spaced yet precisely measured values of $\theta$ are perfectly acceptable. There are currently two techniques for producing intensities at any $\theta$ value in sick, which are described in the following sections.

2.1. N-dimensional Linear Interpolation

The simplest available method is to linearly interpolate between the grid of model spectra. The detailed features of most astrophysical model spectra are unlikely to be well-captured by crude linear interpolation in high dimensions. However, the $N$-dimensional linear interpolation scheme is efficient and may be suitable for many astrophysical instances where only a point-estimate of $\theta$ is required.

This approach uses the Quickhull (Barber et al. 1996) algorithm in SciPy (scipy.interpolate.griddata and scipy.interpolate.LinearNDInterpolator; Jones et al. 2001) to interpolate in $N$-dimensions. The convex hulls required for Quickhull can be globally produced for the entire grid, or for $N_{\text{grid}}$ local points surrounding the initial estimate $\theta_{*\text{,estimate}}$ (see Section 3.1), where $N_{\text{grid}}$ can be specified in the model configuration file. Quickhull relies on Voronoi tessellation, which produces extremely skewed cells when the grid points $(\theta)_{\text{grid}}$ vary significantly in magnitude (e.g., as $T_{\text{eff}}$ does with respect to log g in the examples presented here) and will introduce substantial errors in the interpolation. To minimize these effects, sick automatically scales the grid values $(\theta)_{\text{grid}}$ (in both global and local scenarios) to make $\delta T_{\text{eff, dim}}$ approximately equal (e.g., $\delta T_{\text{eff, scaled}} \approx \delta \log g_{\text{scaled}}$).

The model intensities $I_{\lambda,m}(\theta)$ at wavelengths $\lambda_{m}$ for some arbitrary $\theta$ can be approximated by interpolating from a “nearby” (local or global) grid $(\theta)_{\text{grid}}$

$$I_{\lambda,m}(\theta) \approx I_{\lambda,m}(\theta_{\text{grid}}),$$

where, following the nomenclature in parallel work by Czekala et al. (2014), I denote the symbol $\approx$ as an interpolation operator.

In practice large grids of model spectra are automatically cached using efficient memory-mapping, allowing for the total size of the model grid to far exceed the available random access memory. In other words, grids of model spectra that are hundreds of Gb in size can be efficiently accessed from an external hard disk on any reasonably modern CPU.

2.2. The Cannon

The Cannon (Ness et al. 2015) is a data-driven approach for stellar label determination. The approach makes use of a training set of stars where labels (i.e., $\theta$) have been determined with high fidelity. The normalized, rest-frame pixel information

---

1. However, given the research background of the author, these examples will focus on stellar applications.
2. Libraries of published spectra that are ready-to-use are available for download through the sick download command line tool.
(on a common binning scale) is then used to build a spectral model for a larger sample of stars. Ness et al. (2015) use observed spectra from APOGEE (Majewski et al. 2010) as their training set, and project spectra from the entire survey to efficiently determine stellar parameters. Here I employ The Cannon as a method for producing model intensities within a grid, using synthetic labels.

For the full description of The Cannon, the reader is directed to Ness et al. (2015). Here I describe the algorithm in brief, with a focus on the implementation details in sick. A Cannon model is characterized by a coefficient vector \( \theta_{\lambda,m} \) that allows for the prediction of the intensity \( I_{\lambda,m} \) for a given label vector \( \lambda \), such that at the \( i \)th model pixel with wavelength \( \lambda_{m,i} \):

\[
I_{\lambda,m,i} = g(\lambda_{m,i}, \theta_{\lambda,m,i}) + \text{noise},
\]

(2)

The vector \( \theta_{\lambda,m,i} \) is a set of spectral model coefficients at each \( \lambda_{m,i} \). Thus for the \( i \)th pixel there are a number of coefficients \( \theta_{\lambda,m,i} \) and a scatter term \( s_{\lambda,m,i} \) which we must solve (train). The complexity in \( \theta_{\lambda,m} \) can be adjusted: a linear model is the simplest option, whereas (Ness et al. 2015) employ a quadratic-in-labels model with linear cross-terms. If none is specified, sick will default to a quadratic-in-labels model with linear cross-terms. The label vector \( I_m \) (vector shown),

\[
I_m = [1, \theta_{\lambda,i} - \theta_{\lambda,i}, \theta_{\lambda,j} - \theta_{\lambda,j}, \cdots]
\]

(3)

is generated from a human-readable description of the label vector. The \( \theta_{\lambda,i} \) offsets are taken as the means of the training set. Before using The Cannon, we need to solve for the coefficients \( \{\theta_{\lambda,m}, s_{\lambda,m}\} \) for the label vector \( I_m \) by optimizing the log-likelihood function for each \( \lambda_{m,i} \) pixel,

\[
\ln p(I_{\lambda,m}|\theta_{\lambda,m}, s_{\lambda,m}) = \cdots
\]

\[
-\frac{1}{2} \left( \frac{I_{\lambda,m,i} - \theta_{\lambda,m,i}^T I_{\lambda,m,i}^N S_{\lambda,m,i}^N} {s_{\lambda,m,i}^N + \sigma_{\theta_{\lambda,i}}^2} \right) - \frac{1}{2} \ln(s_{\lambda,m,i}^2 + \sigma_{\theta_{\lambda,i}}^2),
\]

(4)

where \( \sigma_{\theta_{\lambda,i}} \) is the uncertainty in the \( i \)th pixel of the model grid point with parameters \( \theta_{\lambda,i} \). For the remainder of this article I will only use synthesized spectra to produce \( I_{\lambda,m}(\theta) \), and thus \( \sigma_{\theta_{\lambda,i}} \) is zero. However, this term is included in Equation (4) to demonstrate that The Cannon implementation in sick can be trained from existing observed spectra, as originally utilized by Ness et al. (2015).

2.3. Ancillary Effects

With the model intensities \( I_{\lambda,m}(\theta) \) produced, there are still a number of additional phenomena that must be considered before the model can be reliably compared to the data. I will describe the dominant effects in the context of a single observed channel (order/beam/aperture). However, sick allows the user to model any combination of the effects described below, with flexible options for handling multiple channels. For example, although redshift \( z \) is an astrophysical phenomena, if some channels do not benefit from telluric absorption or imprints of Earth-bound rest-frame spectra in their observations, then the wavelength calibration will rely on different arc lines, and consequently be slightly varied in each channel. The user can opt for a single parameter \( z \), or implement separate redshift parameters per channel, with an optional strong joint prior on those parameters.

The model intensities \( I_{\lambda,m}(\theta) \) that have been generated should always be of higher spectral resolution than the data. Therefore it is necessary to convolve the model spectra \( \{\lambda_m, I_{\lambda,m}, S_{\lambda,m}\} \). For these reasons I introduce the spectral resolution \( R = \frac{\lambda}{\Delta \lambda} \) as an additional nuisance parameter. The resolving power can be modeled as a single resolution for all orders, or through additional parameters for each order. For stellar applications, the observed rotational velocity \( v \sin i \) (Gray 2005) can be modeled by an additional kernel, convolved with the \( R \) kernel.

The convolution, resampling, and redshift steps are performed simultaneously in sick. Given the model wavelengths \( \lambda_m \) and observed wavelengths \( \lambda_o \) sick efficiently constructs a sparse matrix \( S \) of size \( N_{\lambda,m} \times N_{\lambda,o} \) such that the expected intensity at an observed pixel \( E_{\lambda,o} \) is given by the dot product of \( I_{\lambda,m}(\theta) \) and \( S \):}

\[
E_{\lambda,o}(\theta_s, \theta_i) = I_{\lambda,m}(\theta_i) \cdot S(v \sin i, z, R, \lambda_m, \lambda_o),
\]

(5)

where \( \theta_i = [v \sin i, z, R, \lambda_m, \lambda_o] \). The vectors \( z \) and \( R \) represent different redshifts and resolving powers in multiple channels.

This data-generation procedure (as illustrated in Figure 1) accounts for all convolutions described above and ensures the total flux is preserved: in notation \( S_{\lambda,m} \sum_{\lambda,o} S_{\lambda,o} = 1 \) for all \( i \). The total width of the convolution kernel (the pixel-convolved line spread function) across the matrix \( S \) for a single channel is dependent on \( v \sin i \) and \( R \) (and subsequently \( \lambda_o \)). The \( (x,y) \) pixel centroid of the convolution kernel along any row or column depends on the model wavelengths \( \lambda_m \) and the redshift \( z \) (through \( \lambda_{shifted} = \lambda_m (1 + z) \)), as well as the wavelengths of the observed pixel \( \lambda_o \). For situations where no kernel convolution is required, a comparable \( S \) matrix is produced to rebin the model to the data, given a redshift \( z \).

The construction of \( S \) constitutes a non-negligible component to the total computational budget for each probability evaluation. For this reason a few (optional) approximations have been implemented to minimize this cost. A least-recently used cache is employed by default to minimize the number of matrix constructions. This uses a small portion of random access memory to retain the \( N_{LRU} \) most common sets of \( \{\theta, S\} \). If \( S(\theta_i + \theta_i) \) is required and \( \theta_i \) is sufficiently small (i.e., below some prescribed tolerance) such that \( S(\theta) \approx S(\theta_i + \theta_i) \), then future calls of \( S \) within the range \( \theta_i \pm \theta_t \) will return the previously calculated matrix \( S \) instead of reconstructing it. The tolerances are configurable, and default to the sub-km s\(^{-1}\) level for \( v \sin i \) and \( z \), and \( \delta R < 1 \). Alternatively, the construction of \( S \) can be completely avoided by approximating the convolution with a single kernel width \( \sigma \) at all \( \lambda_m \), where an interpolation routine is used to calculate the expected intensity \( E_{\lambda,o}(\theta_s, \theta_i) \) at \( \lambda_o \). The extent of approximation that can be afforded will vary on the scientific objectives, but the default behavior in sick is balanced to be computationally efficient, and suitable for most scientific applications.

I have produced the expected intensities \( E_{\lambda,o}(\theta_s, \theta_i) \) for some arbitrary \( \theta_s \) that can fairly represent any rotational broadening of the source, the object’s redshift, as well as the resolving power of the instrument and the location of the CCD pixels. However, these intensities are not representative of real world data. The height (i.e., photon counts) and shape of an observed spectrum is a function of the source magnitude, exposure time, instrument sensitivities, atmospheric conditions, interstellar extinction, and a host of unaddressed effects.
Step 1
Generate model intensities at $\theta_*$ from nearby points $\{\theta_*\}_{\text{grid}}$.

Step 2
Model rotation, redshift, and spectral resolution with matrix
$S(v \sin i, z, R, \lambda, \lambda_0)$:

$$E_{\lambda,0}(\theta_*, \theta_s) = I_{\lambda,0} \cdot S$$

Step 3
Model the continuum with a polynomial function that enters multiplicatively:

$$M_{\lambda,0}(\theta_*, \theta_s, c) = E_{\lambda,0} \cdot C_{\lambda,0}$$

Figure 1. The steps required to produce the model spectrum $M_{\lambda,0}(\theta_*, \theta_s, c)$. This figure is intended to clarify the mathematical nomenclature and visualize the data-generating procedure. Transformations occur from left-to-right, starting from the top row. The spectrum in the top right is a high-resolution ($R \sim 20,000$) portion surrounding H$\alpha$, which is marked in all panels in the rightmost column. The inset axes in the center panel shows a $5 \times 5 \text{Å}$ zoom-in of the convolution kernel. Redder wavelengths have a larger kernel width due to $R$. Positive redshifts move the position of the diagonal convolution to the right. The final model spectrum is shown compared to a low-resolution ($R \sim 2000$) spectrum of a solar-like star with $S/N \sim 30$ per pixel. The additional nuisance parameters $f, p_c, v_0$ only contribute during the calculation of the likelihood $\mathcal{L}$, and do not affect the data generated by the model.

Figure 2. A faux observed spectrum (black) of $R \sim 20,000$ and $S/N$ of $\sim 20$ per pixel, which was used in the toy model test. The recovered maximum a posteriori model spectrum is shown in red.

For these reasons a function $C_{\lambda,0}(c)$ is required to normalize the model to the data. Although the function $C_{\lambda,0}$ incorporates a number of effects (e.g., source blackbody temperature, dust, instrument sensitivities), they are phenomena that usually cannot be separated without additional information, and here I only care about their combined effect. The continuum is modeled as a polynomial that enters
multiplicatively, 
\[ C_{\lambda,0}(e) = \sum_{i=0}^{j} c_i \lambda_i \]
where the maximum polynomial degree is specified by the user. For data spanning multiple channels, I denote \( e \) to represent the continuum coefficients \( e \) in each observed channel. For brevity I define the expected model fluxes at the observed pixels with wavelengths \( \lambda \), as:
\[ M_{\lambda,0}(\theta_s, \theta_e, e) = E_{\lambda,0}(\theta_s, \theta_e) \cdot C_{\lambda,0}(e, \lambda_s). \]  

Thus, the model intensities \( I_{\lambda,m}(\theta_s) \) at model wavelengths \( \lambda_m \) are normalized between 0 and 1 and have no units, as are the expected intensities \( E_{\lambda,0}(\theta_s, \theta_e) \) at observed wavelengths \( \lambda_o \). However, the model fluxes \( M_{\lambda,0}(\theta_s, \theta_e, e) \) at observed wavelengths \( \lambda_s \) have the same “units” as the observations, be it in photon counts (i.e., arbitrary units) or energy flux density (erg s\(^{-1}\) cm\(^{-2}\) Å\(^{-1}\)). That is to say, sick does not require the data to be flux-calibrated.

### 2.4. Underestimated Variance

For data generating models that incorporate an uncertainty in producing each pixel value \( s_{\lambda,m} \) (e.g., Section 2.2), the total variance \( s_{\lambda,o}^2 \) in each pixel is given by:
\[ s_{\lambda,o}^2 = \sigma_{\lambda,o}^2 + C_{\lambda,o}(s_{\lambda,m} \cdot S)^2. \]

The N-dimensional linear interpolation method in Section 2.1 does not allow for a \( s_{\lambda,m} \) term and thus \( C_{\lambda,o}(s_{\lambda,m} \cdot S)^2 \) cancels to zero for the linear interpolation method. However, irrespective of the data-generating method employed, the observed pixel uncertainties \( \sigma_{\lambda,o} \) are usually under-estimated. This may be due to unpropagated uncertainties during data reduction, and/or more commonly a result of untreated covariance between neighboring spectrograph pixels. Ideally the full covariance matrix \( \Sigma \) should be used to fit the data, however, having access to the proper covariance matrix from a data reduction pipeline is not a common scenario.

A robust approach would be to model the neighboring covariance along the diagonal of \( \Sigma \) with a Gaussian process (e.g., see excellent work by Czekala et al. 2014). The downside to employing a Gaussian process to model the global covariance is that matrices of considerable size require regular inversions, potentially adding considerable cost to the computational budget. A simpler approach is to assume the pixel uncertainties \( \sigma_{\lambda,o} \) are systematically underestimated by some fractional amount \( f \). In this scenario the observed variance \( \sigma_{\lambda,o}^2 \) for a given pixel is given by:
\[ s_{\lambda,o}^2 = \sigma_{\lambda,o}^2 + C_{\lambda,o}(s_{\lambda,m} \cdot S)^2 + f^2 M_{\lambda,o}^2. \]

### 2.5. Outliers

I now consider the handling of outliers in the data. These may be in the form of cosmic ray spikes, improper calibration of the data, telluric features, or simply poorly modeled spectral regimes. These pixels can be treated in two ways within sick:
- a Gaussian mixture of the spectrum model and an outlier model, or with semi-constrained \( \sigma \)-clipping at run-time. When a mixture model is employed, the data are fit by the sum of amplitudes \( (1 - p_b) \) and \( p_b \), respectively, where \( b \) represents the “background” outlier model of two distributions: the model fluxes \( M_{\lambda,o} \), and a normal distribution with mean \( C_{\lambda,o} \) and additional variance \( \nu_b \) such that the total variance in a given pixel for the outlier model is \( s_{\lambda,o}^2 + \nu_b \).
- This requires the inclusion of two additional nuisance parameters: \( p_b \) and \( \nu_b \). The prior \( p(p_b) = \mathcal{U}(0, 1) \) is hard-coded in sick. The prior distribution function \( p(\nu_b) \) is similarly fixed, requiring \( \nu_b \) to always be positive (Equation (14)), and as such the outlier distribution will always have a larger variance. Distributions of smaller variance are more informative, so conceptually a fit to the model fluxes \( M_{\lambda,o} \) is generally preferred wherever possible.

### 2.6. Priors

Priors represent our initial knowledge about a particular parameter before looking at the data, and are necessary for any Bayesian analysis. A number of different prior distributions can be specified by the user in the sick model configuration file. The following uninformative prior distributions are assumed (for all channels, where appropriate) unless otherwise specified:
\[ p(\theta_{s,\text{dim}}) = \mathcal{U}(\min[\{(\theta_s)^{\text{grid}}\}], \max[\{(\theta_s)^{\text{grid}}\}]) \]
\[ p(z, \{e\}) = 1 \]
\[ p(\ln f) = \mathcal{U}(-10, 1) \]
\[ p(p_b) = \mathcal{U}(0, 1) \]
\[ p(R, \nu_b) = \begin{cases} 1, & \text{for values greater than zero} \\ 0, & \text{otherwise} \end{cases} \]

There are some subtleties to enforcing \( p(\theta_s) \). A consequence of allowing irregular model grids in the N-dimensional linear interpolation model is that occasionally a spectrum cannot be interpolated for some \( \theta_s \), even if it is bound within \( \min[\{(\theta_s)^{\text{grid}}\}], \max[\{(\theta_s)^{\text{grid}}\}] \) because it will be outside the convex hull. In these cases \( p(\theta_s) = 0 \) and thus \( \ln \mathcal{P} = -\infty \). On the other hand if a Cannon spectral model is used, data can be produced beyond the strict parameter limits that make up the training set. Although this is not a severe restriction on The Cannon, by default sick is cautious and enforces \( p(\theta_s) = 0 \) for points outside the limits of the training set. This option can be disabled by the user.

It is clear that the priors on \( \{e\} \) intuitively should not be uniform. Higher order terms of a polynomial sequence should have much smaller priors, as their absolute magnitudes are expected to be much smaller than lower order terms. In practice the initialization of continuum parameters is normally sufficient such that the default uniform priors on \( \{e\} \) pose no problem, but for the expert user there is clearly room for a formal, well-founded prior to be enforced on \( \{e\} \), which can be enabled in the sick model description. Finally, negative generated model fluxes \( M_{\lambda,o} \) are considered unphysical, and set to non-finite values.

### 2.7. The Likelihood Function

The likelihood function has an additional term if outlier pixels are treated with a mixture model. The parameter \( \Theta \equiv \{\theta_s, v \sin i, R, z, \{e\}, f\} \) describes all parameters in the model. If a Cannon model is used, the implication is that the parameters \( \{\theta_{\lambda,m}, s_{\lambda,m}\} \) have been solved for at each pixel, and are thus already folded into \( I_{\lambda,m}(\theta_s) \) and \( s_{\lambda,o}^2 \). Similarly, due to
the convolution and binning matrix $S$, the model flux at a given pixel $\lambda_{a,i}$ is reliant on knowing the neighboring model and observed wavelengths $\lambda_m$ and $\lambda_a$, but for brevity I do not explicitly specify these terms in Equation (15) below.

I can now describe the frequency (or probability distribution) for the flux at each pixel (of $\{\lambda_{a,i}, F_{a,i}, \sigma_{a,i}\}$) $p(F_{a,i}|\lambda_{a,i}, \sigma_{a,i}, \Theta)$ for the observed data $F_{a,o}$:

$$p(F_{a,i}|\lambda_{a,i}, \sigma_{a,i}, \Theta) = \frac{1}{2\pi \sigma_{a,i}^2} \exp \left( - \frac{(F_{a,i} - M_{a,i})^2}{2\sigma_{a,i}^2} \right)$$  \hspace{1cm} (15)

where $s_{a,o}^2$ was defined in Equation (9). If the outlier pixels are also being modeled then the probability distribution for $F_{a,o}$ becomes a mixture of two models:

$$p(F_{a,i}|\lambda_{a,i}, \sigma_{a,i}, \Theta, p_b, v_b) = \ldots$$

$$\left(1 - p_b\right) \times p(F_{a,i}|\lambda_{a,i}, \sigma_{a,i}, \Theta) + \ldots$$

$$p_b \times p_{\text{background}}(F_{a,i}|\lambda_{a,i}, \sigma_{a,i}, \Theta, v_b).$$  \hspace{1cm} (16)

Where $p_{\text{background}}$ is defined as:

$$p_{\text{background}}(F_{a,i}|\lambda_{a,i}, \sigma_{a,i}, \Theta, v_b) = \ldots$$

$$\frac{1}{2\pi (s_{a,o}^2 + v_b)} \exp \left( - \frac{(F_{a,i} - c_{a,i})^2}{2(s_{a,o}^2 + v_b)} \right).$$  \hspace{1cm} (17)

3. METHODOLOGY

sick aims to be flexible to achieve different scientific objectives, depending on the data volume and the computing resources available. Generally, when I have acquired some data, I seek either a coarse guess of the model parameters $\Theta$, a numerically optimized point estimate of $\Theta$ (without uncertainties), or full sampling of the posterior probability distribution. sick has three primary analysis functions to suit these scenarios. Below I list the abridged sick command line usage for each situation, as well as a brief description:

1. sick estimate <model><data>

An initial estimate of the model parameters $\Theta_{\text{estimate}}$ is obtained by cross-correlating a pseudo-normalized copy of the data against the entire model grid (or some subset thereof; Section 3.1). The nearest neighbor $\theta^*_{\text{nearest}}$ is returned, and $\{c\}$ and $z$ are estimated from $\theta^*_{\text{nearest}}$.

2. sick optimize<model><data>

Numerical optimization of $-\ln P$ begins from $\Theta_{\text{estimate}}$, unless an initial guess is provided. If no minimization algorithm is selected, a scaled and bounded version of the Broyden, Fletcher, Goldfarb, and Shanno (BFGS; Section 3.2) algorithm is used.

3. sick infer<model><data>

Markov Chain Monte Carlo (MCMC) sampling begins from the numerically optimized point $\Theta_{\text{estimate}}$. Sampling occurs for at least 2000 steps with 200 walkers (4 $\times$ $10^5$ probability evaluations), with convergence automatically determined from the autocorrelation functions, unless conflicting sampling requirements have been provided by the (presumably expert) user.

3.1. Initial Estimate

I require a good initial estimate of the model parameters $\Theta_{\text{estimate}}$. Initially sick fits each observed channel with a polynomial (with degree set by the user in the model configuration file), discards pixels that deviate by more than 4$\sigma$, and repeats the fit. A copy of the data is divided by the fitted continuum, yielding a “pseudo-normalized” spectrum (e.g., the spectrum is “normalized” without any consideration of strong molecular bands or continuous opacities depressing the entire spectrum). This “pseudo-normalized” spectrum is then cross-correlated against the entire model grid, or $N_{\text{grid-estimate}}$ equispaced points across $\{\theta^*_\text{grid}\}$.

The relative peak of the cross-correlation function (CCF) $F_{\text{ccf}}$ provides a reliable metric of similarity between two spectra, thereby providing a cheap estimate of the model parameters $\theta_*$. Given the nearest-neighbor guess of $\theta_*$ and the redshift $z$, the optimal continuum coefficients $c$ can then be calculated algebraically after pseudo-sampling (i.e., interpolating) $I_{\lambda,m(1+z)}$ on to $\lambda_o$.

$$F_{\lambda,o} \approx I_{\lambda,m(1+z)}(\theta_*) \cdot C_{\lambda,o}(c, \lambda_o).$$  \hspace{1cm} (18)

When multiple channels are present (and the redshift is being modeled separately in each channel), an estimate of the closest model parameters $\theta_*$ is provided by each channel. In this scenario sick calculates the optimal continuum coefficients $c$ in each channel, for each unique value of the set $\{\theta_*\}$ returned from the peaks of the CCFs.

The $\chi^2$ difference (calculated using all channels) between the approximate model (Equation (18)) and the data is calculated for each entry in $\theta_*$. The point with the lowest total $\chi^2$ value is taken as the initial estimate of $\theta_*$ and its corresponding redshift(s) $z$ and continuum coefficients $\{c\})$. The spectral resolution(s) $R$ are estimated from the wavelength spacing $\lambda$ in each channel. The fraction of underestimated variance is assumed to be high ($\ln f = 0.5$), and the outlier fraction is initially estimated to be small (1%), with the outlier variance assumed to be comparable to the observed pixel variance $v_b = (\sigma^2)_{a,i}^2$.

3.2. Optimization

The model parameters $\Theta$ are then numerically optimized, using $\Theta_{\text{estimate}}$ as the starting point. I numerically optimize the parameters $\Theta$ by minimizing the negative log-probability $-\ln P$. A number of suitable minimization algorithms are available in sick through the SciPy (Jones et al. 2001) optimization module:

1. BFGS (Byrd et al. 1995; Zhu et al. 1997; Nocedal & Wright 2006; Morales & Nocedal 2011) default.
2. Modified Powell’s method (Powell 1964; Press et al. 2002).
3. Nonlinear conjugate gradient method (Nocedal & Wright 2006).
4. Truncated Newton conjugate-gradient method (Nash 1984; Nocedal & Wright 2006).
5. Nelder–Mead (Nelder & Mead 1965).

If bounded information is available (e.g., boundaries of $\theta^*_{\text{grid}}$) or limits from uniform priors, sick will use constrained implementations of the algorithms above, where they exist. For algorithms that utilize a single parameter scaling factor for gauging convergence (e.g., factr in BFGS), sick

---

\(^a\) This procedure constitutes the antithesis of an earlier footnote, but is only being performed to facilitate a cheap comparison between the data and all possible models.
automatically scales $\Theta$ to place the parameters in the same order of magnitude. Tunable convergence parameters for each optimization algorithm are also configurable through the model configuration file. However, the default options in sick should be suitable for most purposes. Model parameters can also be optionally fixed during the optimization process. For example, if only a point estimate of $\Theta$ is required (e.g., no MCMC sampling) and the procedure in Section 3.1 provides a reliable measure of $z$, one might choose to keep $z$ fixed for the optimization process and solve for the remaining $\Theta$.

3.3. Monte-Carlo Markov Chain Sampling

The aforementioned steps efficiently provide an accurate point estimate of the optimal parameters $\Theta_{\text{optimized}}$. However, they do not provide a measure of uncertainty on $\Theta$, which is usually more important than a single value. sick employs the affine-invariant ensemble Metropolis–Hastings sampler proposed by Goodman & Weare (2010) and implemented by Foreman-Mackey et al. (2013). The initial distributions of $\Theta$ (hereafter called the initial state $\alpha$) are drawn from a small multi-dimensional ball around the optimized parameters $\Theta_{\text{optimized}}$.

Before the posterior distributions of $\Theta$ can be properly sampled, the MCMC chains must be thermalized (i.e., “burnt in”) from the initial proposal $\alpha$ to the equilibrium distribution $\pi$. It can be shown (e.g., Chung 2006) that a Markov chain will converge to $\pi$ as $t \to \infty$. However the initial transit period from $\alpha$ to $\pi$ must be discarded; our posterior $p(\Theta)$ should not depend on the initial distribution $\alpha$. Ensuring a Markov chain converges to equilibrium in finite time is a fundamental topic in statistics.

In an attempt to make sick easy to use, some default behavior has been introduced to routinely evaluate whether convergence has probably been achieved. Consider the transit from the $\alpha$ to $\pi$ distributions. Each successive state $(\Theta_h, \Theta_{h+1}, \cdots)$ of the Markov chains is correlated. In other words, each state depends slightly on the previous state. This is the exact opposite of what I actually want to achieve, as I seek $\pi$ to be effectively independent of the initial distribution $\alpha$. Thus the auto-correlation between successive states is informative of—among other things—whether the Markov chains are near $\alpha$ or $\pi$. If one considers an observable $f$ (e.g., any parameter in $\Theta$), I can estimate its mean,

$$\langle f \rangle = \int f(\phi)p(\phi)d\phi,$$

where $p(\phi)$ is the probability density function. The unnormalized autocorrelation function $C_f$ between two successive states $t - 1$ and $t$ is then given by:

$$C_f(t) \equiv \langle f_{t-1}f_t \rangle - \mu_f^2 \tag{20}$$

What is of most interest is the normalized autocorrelation function $\rho_f(t)$ at a given time step $t$, as normalized to the initial ($t=0$) state $\alpha$. That is to say, the normalized autocorrelation function $\rho_f(t)$ provides a measure of how correlated a value $f$ (at time $t$) is to the initial distribution $\alpha$ (when $t=0$):

$$\rho_f(t) \equiv C_f(t) / C_f(0). \tag{22}$$

The normalized autocorrelation function will decay exponentially for large $t$ as the states move closer to the equilibrium distribution $\pi$. In other words, at some large $t$ (which is unknown a priori) the state $\Theta_t$ will be independent, or negligibly dependent on the initial state $\alpha$. Therefore it is important to know how quickly the exponential decay of $\rho_f$ is, such that I can identify where the chains have properly thermalized and settled in to $\sim\pi$. The exponential autocorrelation time $\tau_{\exp,f}$ provides the measure of decay and is defined as,

$$\tau_{\exp,f} = \lim_{t \to \infty} \sup \frac{t}{-\ln|\rho_f(t)|}. \tag{23}$$

where $\tau_{\exp}$ provides the relaxation time of the system, as given by which parameter $f$ is moving slowest from $\alpha$ to $\pi$:

$$\tau_{\exp} = \sup_f \tau_{\exp,f}. \tag{24}$$

Therefore, if $t$ is large enough, the burn-in period can be estimated as a $\sim$ few multiples of the relaxation time of the system $\tau_{\exp}$. However there is some ambiguity about when $\tau_{\exp}$ can be calculated, because for short chains $\tau_{\exp}$ is forced to be shorter than the chain length by construction. For this reason we must sample a “sufficient” number of times before estimating the exponential autocorrelation time. Adopting conservative behavior by default, sick will initialize 200 walkers and run until $\tau_{\exp} < 1/64$ the chain length. In practice this may be overkill, but this behavior can be changed by the user.

I estimate $\tau_{\exp}$ by first calculating the normalized autocorrelation function $\rho_f$ for each parameter, using the mean position of all walkers at any time $t$. For any time $t$ I take the maximum absolute $\rho_f$ for any parameter $f$: $\rho_{\max} = \sup_f |\rho_f|$. Thus, $\rho_{\max}$ gives the upper limit of autocorrelation in any parameter $f$ at a given time $t$. Finally, I fit an exponential function of the form $\exp(-\frac{1}{\tau_{\exp}})$ to the profile $(t, \rho_{\max})$ by least-squares minimization in order to estimate $\tau_{\exp}$.

Given the estimate of $\tau_{\exp}$ I discard the first $3 \times \tau_{\exp}$ MCMC steps as the thermalization phase. Using the remaining samples, I next calculate the integrated autocorrelation time $\tau_{\text{int}}$ (using the emcee.autocorr.integrated_time function), which is distinct from the exponential autocorrelation time discussed above. The integrated autocorrelation time is defined as

$$\tau_{\text{int},f} = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_f(t). \tag{25}$$

Because the integrated autocorrelation time $\tau_{\text{int},f}$ is calculated only on samples after $3 \times \tau_{\exp}$ (i.e., after thermalization has occurred), it provides a measure of the
statistical error in the Monte Carlo samples of parameter \( \langle f \rangle \), and a means of determining the number of effective independent samples of \( f \),

\[
N_{\text{eff},f} \approx \frac{N_{\text{steps}}}{2\tau_{\text{int},f}},
\]

where \( N_{\text{steps}} \) is the number of production (post-thermalization) MCMC steps. The reader is referred to the excellent notes by A. D. Sokal\(^5\) for more details (and clear derivations) of sample estimators and autocorrelation times.

After the thermalization regime has been identified, the default constraint for convergence in \texttt{sick} is to have more than 100 effective independent samples in every parameter \( f \). If this heuristic is not met after the first 2000 steps, \texttt{sick} will calculate \( \tau_{\text{exp}}, \tau_{\text{int}}, \) and \( N_{\text{eff}} \) every 1000 MCMC steps thereafter. Expert users can modify this behavior by disabling convergence checking completely (with the \texttt{auto\_convergence} setting) in lieu of specifying the number of iterations to burn and sample, or by altering the following convergence criteria settings: \texttt{n\_tau\_exp\_as\_burn\_in, minimum\_effective\_independent\_samples, check\_convergence\_frequency, and minimum\_samples}, which will specify the minimum number of MCMC steps before evaluating convergence.

Once sampling is complete, \texttt{sick} generates figures showing the normalized auto-correlation \( \rho_g \) in each parameter, the mean acceptance fraction at each step, all of the sampled \( \Theta \) parameters in each chain, corner plots showing marginalized posteriors for \( \Theta \) and \( \Theta_\ast \), as well as projection (spectrum) plots that illustrate the quality of fit to the data. Furthermore, the chains and final state of every MCMC is saved by \texttt{sick}, allowing users to resume their analysis from the most recent state, calculate additional sample estimators, or to produce supplementary post-processing figures.

4. TOY MODEL

A straightforward test of the probabilistic framework described above is to produce a noisy faux observation, and infer the model parameters \( \Theta \), given the faux data. The AMBRE public spectral library (de Laverny et al. 2012) has been employed as the grid of model intensities. I used \( \sim4000 \) points \( \{ \theta_\ast \}^{\text{grid}} \) in the range \( 4000 < T_{\text{eff}} < 8000, \ 0 < \log g < 5, \ -3 < [\text{Fe/H}] < 0.5 \) and \( -0.2 < [\alpha/\text{Fe}] < 0.8 \) to train a Cannon quadratic-in-labels model with linear cross-terms. Stellar parameters \( \{ T_{\text{eff}}, \log g, [\text{Fe/H}], [\alpha/\text{Fe}] \} \) of a solar-like star \( \theta_\ast = [5841, 4.41, -0.03, -0.01] \) were chosen, and model intensities \( \lambda_{\text{int}} \) were generated between \( \lambda_{\text{m}} = [8450, 8900] \) Å.

The wavelength range is quite common to many surveys and instruments: RAVE, FLAMES/GIRAFFE (HR21), \textit{Gaia} RVS, AAOmega (1700D). For this test, the spectral resolution is most comparable to FLAMES/GIRAFFE. For the purposes of producing a faux observation, I disregard the uncertainties in producing the model intensities \( \sigma_{\lambda,m} \). In other words, \( \sigma_{\lambda,m} \) does not contribute to the quoted noise in \( \sigma_{\lambda,o} \). However when generating model intensities for probability evaluation at runtime, \( \sigma_{\lambda,m} \) enters into the likelihood function as per Equation (9).

\(^5\) http://www.stat.unc.edu/faculty/cj/Sokal.pdf

A number of transformations were then applied to the intensities. The spectra were convolved to a resolving power of \( R \sim 20,000 \) and redshifted by a random velocity drawn from \( N(0, 300) \) km s\(^{-1}\) before the data were binned onto a uniform spacing of 0.1 Å. A second-order polynomial was used to represent the continuum, and noise was added to replicate a signal-to-noise ratio (S/N) of \( \sim20 \) per pixel.

The model included the parameters \( T_{\text{eff}}, \log g, [\text{Fe/H}], [\alpha/\text{Fe}], \zeta, R, c_0, c_1, c_2, \) and \( \ln f \). Two hundred walkers sampled for 5000 steps to thermalize the sampler. In practice, this is more samples than what was necessary for this test, as evidenced by the normalized auto-correlation function \( \rho_g(t) \) (Figure 3). The chains were reset and another 5000 MCMC steps were performed to sample the posterior distribution. The posterior probability distributions of \( \theta_\ast \) are shown in Figure 4, where blue marks the parameters used to generate the data. Table 1 lists the maximum a posteriori values of \( \Theta \) (which is projected to the data in Figure 2) and the 16th and 84th percentiles of the \( \Theta \) distributions. It is clear from Figure 4 and Table 1 that the model recovers the data-generating values \( \Theta \) very well.

5. UTILITY: ATOMIC DIFFUSION IN M67

Although \texttt{sick} can be used to estimate a (nearest-neighbor or numerically optimized) point estimate, I have spent considerable effort describing the dominant conceivable phenomena that may affect the data, and outlined how to incorporate those nuisance effects into a scalar-justified model. Given the additional (often considerable) computational cost implied by MCMC to marginalize over these nuisance parameters, it is reasonable to ask “Why bother?” Does the introduction and marginalization of nuisance parameters actually improve our inferences on astrophysical parameters \( \theta_\ast \)? In other words, if a full sampling of the posterior and marginalization of nuisance parameters does not provide additional scientific information, is it pragmatic to perform MCMC? The computational cost may not be warranted.

Here I present a suitable application that demonstrates that there is additional information in existing public spectra which has not been fully exploited. M67 is a nearby (~800–900 pc;
sandquist 2004; sarajedini et al. 2009; yakut et al. 2009; majaess et al. 2011) open cluster with a near-solar metallicity: [fe/h] = −0.04 to +0.03 (hobbs & thorburn 1991; tautvaisiene et al. 2000; yong et al. 2005; randich et al. 2007; pace et al. 2008; pasquini et al. 2008; önehag et al. 2011). the age of the cluster is comparable to the sun (3.5–4.8 gyr), and represents an excellent test-bed for stellar evolution and diffusive convection at solar metallicity (e.g., vandenberg et al. 2008).

candidate stars in m67 were observed on the australian astronomical telescope with the aaoomega instrument in may 2011. the 1700d grating was used in the red arm, which gives comparable wavelength coverage to the toy model in section 4, but at a lower resolution of r ~ 10,000. i convolved the ambre spectral library (de laverny et al. 2012) to this spectral resolution while keeping the high-resolution sampling. a cannon model with label vector [1, teff, log g, [fe/h], [α/fe], teff · log g, teff · [fe/h], teff · [α/fe], log g · [α/fe], [fe/h] · [α/fe], teff · log g · [fe/h], [α/fe]] was trained across the region 4000 < teff < 7000, 1.0 < log g < 5.0, −2.5 < [fe/h] < 0.5, and −0.4 < [α/fe] < 0.4 (4202 points). once the model was trained, i used the “sick infer” command line (with prescribed “burn” and “sample” values, see below) to infer astrophysical parameters given the 1700d spectra. as described in section 3, the− ln p was numerically optimized from a nearest-neighbor point estimate.
of the parameters $\Theta$, before performing MCMC sampling with 200 walkers for 2000 steps in thermalization and production. The model parameters were $\Theta = [T_{\text{eff}}, \log g, [\text{Fe/H}], [\alpha/\text{Fe}], z, \mathcal{R}, \ln f, c_0, c_1, c_2, p_b, v_b]$. Cluster members were unambiguously identified from their inferred redshifts $z$. Suspected spectroscopic binaries (due to significant line broadening or resolved double-peaks in their spectrum) were discarded. The distilled sample includes 24 members, which are shown in Figure 5, with a 4.5 Gyr solar-metallicity PARSEC (Bressan et al. 2012) isochrone to guide the eye. The sample consists of predominantly turn-off and sub-giant stars. The discrepancy with the isochrone at the giant branch is a noticeable, well-studied effect (e.g., Vandenberg 1983). Indeed, the position of the red giant branch is a sensitive function of the (invoked) $\alpha$ parameter in mixing length theory. For this reason M67 has been a useful boundary condition for testing blanketed model atmospheres (e.g., Vandenberg et al. 2008). Without this condition, the position of the red giant branch in 4.5 Gyr solar isochrones tends redwards (e.g., cooler temperatures), to the same degree that I find in Figure 5. The uncertainties are sufficiently large that I cannot precisely distinguish between stars that have passed the turn-off point, but the overall agreement with the sequence is satisfactory.

If I assume that the underlying metallicity distribution of M67 is a Gaussian, and account for the uncertainties in [Fe/H] for each star, from all cluster members I find the metallicity of M67 to be $[\text{Fe/H}] = -0.06$. This measurement is in reasonable agreement with existing studies that place the metallicity of M67 between $-0.04$ and $+0.03$. However, Figure 5 shows the sample is two-thirds dominated by turn-off stars. In Figure 6 I show the inferred maximum a posteriori metallicity for all M67 stars, binned to 0.05 dex increments (i.e., roughly equivalent to the uncertainties in individual measurements). The turn-off and sub-giant stars are colored in the same way as Figure 5, and the hatching represents an overlap between the two distributions. As a whole, the turn-off stars are more metal-poor than the sub-giant stars. If the uncertainties in [Fe/H] for each star are considered, and assume the underlying metallicity distribution in the sub-giant and turn-off sample are normally distributed, I find the metallicity of turn-off stars to be $[\text{Fe/H}] = -0.07$, and $[\text{Fe/H}] = -0.02$ for the sub-giant stars.

The difference in metallicities between sub-giant and turn-off stars in M67 is not a new result. Önehag et al. (2014) performed a differential study of sub-giant and turn-off stars in M67 with respect to the solar twin M67-1194. Their data were obtained from the UVES instrument on the VLT, providing $\mathcal{R} \sim 47,000$ and S/N $\sim 150$ per (binned) pixel. This data

| Parameter | Description | Data-generating Value | MAP Value |
|-----------|-------------|-----------------------|-----------|
| $T_{\text{eff}}$ | Effective photospheric temperature (K) | 5841 | 5825$^{+38}_{-38}$ |
| $\log g$ | Surface gravity | 4.41 | 4.41$^{+0.05}_{-0.05}$ |
| [Fe/H] | Metallicity | $-0.03$ | $-0.04^{+0.03}_{-0.03}$ |
| $[\alpha/\text{Fe}]$ | $\alpha$-element enhancement | $+0.01$ | $+0.02^{+0.02}_{-0.02}$ |
| $z \cdot c$ | Redshift/Doppler shift (km s$^{-1}$) | 43.1 | 43.7$^{+0.1}_{-0.4}$ |
| $\mathcal{R}$ | Spectral resolution | 20000 | 21648$^{+2346}_{-2335}$ |
| $c_0$ | Continuum polynomial coefficient | $-8000$ | $-9050^{+2338}_{-2338}$ |
| $c_1$ | Continuum polynomial coefficient | 1.14 | 1.39$^{+0.54}_{-0.54}$ |
| $c_2$ | Continuum polynomial coefficient ($\times 10^{-5}$) | $-5$ | $-1.42^{+2.11}_{-2.14}$ |
| $\ln f$ | Logarithm of fractionally underestimated variance | ... | $-8.93^{+0.85}_{-0.71}$ |

Figure 5. Inferred effective temperatures and surface gravities of confirmed cluster members of M67. A single-mass 4.5 Gyr PARSEC (Bressan et al. 2012) isochrone of solar metallicity is shown to guide the eye. The discrepancy in the position of the red giant branch is a known astrophysical phenomena (see the text). When accounted for, there is good overall agreement with the sequence. Samples are labeled as turn-off or sub-giant stars.

Figure 6. A histogram showing the maximum a posteriori metallicity for confirmed M67 stars, colored by their evolutionary stage as per Figure 5. As a whole, the sub-giant stars exhibit slightly ($+0.05$ dex, when individual uncertainties in [Fe/H] are considered) higher metallicities than the turn-off stars, a result previously identified from high-resolution spectra by (Önehag et al. 2014) and attributed to atomic diffusion.
quality permitted the determination of individual chemical abundances in a strictly differential sense. It is important to remember that the metallicity difference I find is probably a convolution of multiple elements within this wavelength range (Fe, Ti, and Ni). With this caveat in mind—and although Önehag et al. (2014) cover a slightly different temperature range—the effect I find is the same: heavy-element abundances (including Fe) in M67 are found to be reduced in the hotter stars and dwarfs by typically $\leq 0.05$ dex, as compared to the abundances of the sub-giants. Thus, with an objective characterization of (most) dominant phenomena that will affect the observations, subtle astrophysical phenomena can be inferred from lower-resolution spectra than what is typically considered. This example application shows, to a large extent, that existing stellar spectroscopic data are sufficiently high-quality that the standard of our results will be dominated by our analysis methods. For this reason, a move toward generative models in stellar spectroscopy is essential.

6. DISCUSSION

Here I discuss some future applications that for sick, are an important potential caveat, and outline planned near-term improvements for the code.

6.1. Further Applications

As previously discussed, sick is agnostic about wavelength coverage, resolving power, or binning of the observed data. This generality allows for an extremely high-resolution grid to be used for many applications of lower resolutions, as long as the wavelengths are covered. For example, the same high-resolution library (observed or synthetic) can be used for surveys of high-resolution (e.g., APOGEE), low-resolution (e.g., SEGUE, Gaia RVS or Gaia BP/RP$^7$—where $R \sim 120$), placing stars from all surveys on the same, self-consistent scale.

The flexibility in methodology (e.g., estimate, optimize, infer) also allows for quick analyses over extremely large data sets (e.g., LAMOST) to identify superlateral objects with high scientific impact (e.g., ultra metal-poor or high-velocity stars). Similarly, large collections of point estimates can be sufficient to identify and quantify substructure in the Milky Way halo. Alternatively, sampling posterior probability distributions for a reasonable sample of cluster stars may reveal subtle abundance variations, helping to untangle the “multiple population” scenario (e.g., see Carretta et al. 2009, and references therein) or understand the effects of atomic diffusion in clusters of different ages and metallicities.

6.2. Caveats

Although there are substantial scientific applications for the tool presented here, there is an obvious caveat that requires attention. The examples here have focused on producing model intensities $I_{\lambda,m}$ from at most four-dimensions. As the number of dimensions increases—for example, to include individual chemical abundances—the curse of dimensionality will quickly become relevant. The computational complexity in producing $I_{\lambda,m}$ scales quickly, such that a linear interpolation model will become absolutely unsuitable in higher dimensions. On the other hand, there are some tricks that can be introduced for a Cannon-like approach in high-dimensionality for chemical abundances (e.g., see A. R. Casey et al. 2016, in preparation; M. Ness et al. 2016, in preparation). Alternatively, compromising two suitable models may be a reasonable scenario: one for the determination of stellar parameters, and another that produces spectra of individual elemental abundances (by some means, Cannon-like or synthesized) for some small wavelength region and stellar parameters $\theta_i$. The abundance of an individual element can be marginalized over all possible $\theta_i$.

The implied assumption here is that the individual chemical abundances do not have a substantial effect on the overall stellar parameters. It is tempting to assert that this is an unwarranted generalization. However, this is an implied assumption used in the production of model photosphere, since the photospheres themselves are calculated with a given chemical composition of individual elements. Although there are potential ways to deal with higher dimensionality (in stellar applications), extreme care must be made in scaling the intensity-generating methods described here.

There are a number of caveats that users must be cognizant of even for models with low-dimensionality. These apply to the data and the model employed. For example, this framework is most suitable for one-dimensional extracted spectra; it is beyond the scope of this work to forward model two-dimensional images. Likewise, sick may be sub-optimal for echelle spectra with many orders due to the high number of $c$ coefficients that are physically related.

Even within a single order, there are strong assumptions made about the line spread function. It is assumed that the resolution scales (at most) linearly with wavelength, which can be shown to be knowingly incorrect for most spectrographs. The effect of this assumption will usually be small (a small second order resolution term is probably warranted), but it should be known.

In the work presented here, flux noise is assumed to be Gaussian. Similarly the noise can only assumed to be underestimated (not over-estimated), and no framework has been presented to account for correlated noise between neighboring pixels (e.g., Czekala et al. 2014). It is also important to note that the treatment of outliers in this work may be overkill: rejecting highly discrepant pixels may be sufficient. In short, the user should be extremely familiar with the model description they set out for the data, and the limitations thereof. Improvements to the code that help resolve these existing caveats are welcomed in the form of pull requests through GitHub.\footnote{Indeed, sick performed excellently in a blind test during the 3rd Gaia Challenge Workshop, slightly outperforming the current BP/RP analysis method for metal-poor stars: http://astrowiki.ph.surrey.ac.uk/dokuwiki/doku.php?id=tests:astopars:challenge3}

7. CONCLUSION

I have presented a flexible probabilistic code to forward model spectroscopic data. The generative model approach described here has a number of advantages over previously published techniques. Preparatory and subjective decisions (e.g., redshift and placement of continuum) are objectively treated within a scalar-justified mathematical model, allowing for a credible assessment of uncertainties in astrophysical parameters. Almost all previously published techniques have treated these processes separately, thereby increasing biases in their results and generally mis-characterizing the uncertainties in astrophysical parameters.\footnote{github.com/andycasey/sick}
The simultaneous incorporation of continuum, redshift, convolution, and resampling leads to remarkable improvements in both accuracy and precision. While the examples presented here have focused on stellar spectra, the code is ambivalent about what the astrophysical parameters describe: the framework can be easily used for any kind of quantitative astrophysical process. The code is MIT-licensed (freely distributable, open source) and has an extensive automatic testing suite. Complete documentation is available online, which includes a number of additional examples and tutorials on analyzing data from well-known stellar surveys (e.g., SEGUE, LAMOST, APOGEE).

Great effort has been made to ensure the code is easy to use, allowing users to obtain precise inferences with little effort. I strongly encourage the use of this software for existing and future spectral data. With the sheer volume of high-quality of spectra available, astronomers must begin to adopt objective, generative models for their data. Subtle astrophysical processes can only be discovered and understood with the proper characterization of uncertainties afforded by generative models.

I am pleased to thank Matt Auger, Sergey Koposov, Melissa Ness, Jarryd Page, Jason Sanders, and Kevin Schlaufman. I am particularly indebted to Brian Schmidt for introducing me to Bayesian statistics and generative models (far too late, I might add), and to the referees for their constructive and direct criticism, which improved the methodology of the code and overall clarity of this paper. This research has made extensive use of NASA's Astrophysics Data System Bibliographic Services, the Coveralls continuous integration service, GitHub, and the trianglepy code (Foreman-Mackey et al. 2014). The author recognizes support through the European Research Council grant 320360: The Gaia-ESO Milky Way Survey. The source code for this research is distributed using cgit, and is hosted online at GitHub. Suggestions for improvements or unexpected behavior can be reported through GitHub issues, and code contributions are welcomed in the form of pull requests.

REFERENCES

Allende Prieto, C., Sivarani, T., Beers, T. C., et al. 2008, AJ, 136, 2070
Barber, C. B., Dobkin, D. P., & Huhdanpaa, H. 1996, ACM Transactions on Mathematical Software, 22, 469
Blondin, S., & Tonry, J. L. 2007, ApJ, 666, 1024
Bressan, A., Marigo, P., Girardi, L., et al. 2012, MNRAS, 427, 127
Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. 1995, SIAM J. Sci. Comput., 16, 1190
Carretta, E., Bragaglia, A., Gratton, R. G., et al. 2009, A&A, 505, 117
Chang, K. L. 2006, Markov Chains With Stationary Transition Probabilities (Berlin: Springer-Verlag)
Cui, X.-Q., Zhao, Y.-H., Chu, Y.-Q., et al. 2012, RAA, 12, 1197
Czekala, I., Andrews, S. M., Mandel, K. S., Hogg, D. W., & Green, G. M. 2014, arXiv:1412.5177
de Laverny, P., Recio-Blanco, A., Worley, C. C., & Plez, B. 2012, A&A, 544, A126
De Silva, G. M., Freeman, K. C., Bland-Hawthorn, J., et al. 2015, MNRAS, 449, 2604
Drinkwater, M. J., Jurek, R. J., Blake, C., et al. 2010, MNRAS, 401, 1429
Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306
Foreman-Mackey, D., Price-Whelan, A., Ryan, G., et al. 2014, trianglepy v0.1.1, Zenodo, doi:10.5281/zenodo.11020
Gilmore, G., Randich, S., Asplund, M., et al. 2012, Msngr, 147, 25
Goodman, J., & Weare, J. 2010, Comm. Appl. Math. Comp. Sci., 5, 65
Gray, D. 2005, The Observation and Analysis of Stellar Photospheres (Cambridge: Cambridge Univ. Press)
Hobbs, L. M., & Thorburn, J. A. 1991, AI, 102, 1070
Husser, T.-O., Wende-von Berg, S., Dreizler, S., et al. 2013, A&A, 553, A6
Jones, E., Oliphant, T., Peterson, P., et al. 2001, SciPy: Open Source Scientific Tools for Python. [Online; accessed 2014-07-14]
Kerzendorf, W. E., Yong, D., Schmidt, B. P., et al. 2013, ApJ, 774, 99
Kordopatis, G., Gilmore, G., Steinmetz, M., et al. 2013, AJ, 146, 134
Le Borgne, D., Rocca-Volmerange, B., Prugniel, P., et al. 2004, A&A, 425, 881
Lee, Y. S., Beers, T. C., Sivarani, T., et al. 2008, AJ, 136, 2022
Majaaes, D. J., Turner, D. G., Lane, D. J., & Kraici, T. 2011, JAVSO, 39, 219
Majewski, S. R., Wilson, J. C., Hearty, F., Schiavon, R. R., & Skrutskie, M. F. 2010, in IAU Symp. 265, ed. K. Cunha, M. Spitce, & B. Barbry (Cambridge: Cambridge Univ. Press), 480
Morales, J. L., & Nocedal, J. 2011, ACM Trans. Math. Softw., 38, 7
Nash, S. G. 1984, SINJ, 21, 770
Nelder, J. A., & Mead, R. 1965, CompJ, 7, 308
Ness, M., Hogg, D. W., Rix, H.-W., Ho, A., & Zasowski, G. 2015, arXiv:1501.07604
Nocedal, J., & Wright, S. J. 2006, Numerical Optimization (2nd ed.; New York: Springer)
Onehag, A., Gustafsson, B., & Korn, A. 2014, A&A, 562, A102
Onehag, A., Korn, A., Gustafsson, B., Stempels, E., & Vandenberg, D. A. 2011, A&A, 528, A85
Pace, G., Pasquini, L., & François, P. 2008, A&A, 489, 403
Palacios, A., Gebran, M., Jesselin, e., et al. 2010, A&A, 516, A13
Pasquini, L., Biazzo, K., Bonifacio, P., Randich, S., & Bedin, L. R. 2008, A&A, 489, 677
Powell, M. J. D. 1964, CompJ, 7, 155
Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 2002, Numerical Recipes in C++: The Art of Scientific Computing (2nd ed.; Cambridge: Cambridge Univ. Press)
Randich, S., Primas, F., Pasquini, L., Sestito, P., & Pallavicini, R. 2007, A&A, 469, 163
Recio-Blanco, A., Bijouj, A., & de Laverny, P. 2006, MNRAS, 370, 141
Sandquist, E. L. 2004, MNRAS, 347, 101
Sanjó Jalili, A., Dotter, A., & Kirkpatrick, A. 2009, ApJ, 698, 1872
Schlegel, D. J., Blanton, M., Eisenstein, D., et al. 2007, BAAS, 39, 132
Smiljanic, R., Korn, A. J., Bergemann, M., et al. 2014, A&A, 570, A122
Sousa, S. G., Alapini, A., Isaiachtin, G., & Santos, N. C. 2010, A&A, 512, A13
Steinmetz, M., Zwitter, T., Siebert, A., et al. 2006, AJ, 132, 1645
Stetson, P. B., & Pancino, E. 2008, PASP, 120, 1332
Tartviäsiene, G., Edvardsson, B., Tuominen, I., & Ilyin, I. 2000, A&A, 360, 499
Torres, G., Fischer, D. A., Sozzetti, A., et al. 2012, ApJ, 757, 161
Valenti, J. A., & Piskunov, N. 1996, A&As, 118, 595
Vandenberg, D. A. 1983, ApJS, 51, 29
Vandenberg, D. A., Gustafsson, B., Edvardsson, B., Eriksson, K., & Fergusson, J. W. 2008, MmSAI, 79, 759
Yakut, K., Zima, W., Kalomeni, B., et al. 2009, A&A, 503, 165
Yanny, B., Rockosi, C., Newberg, H. J., et al. 2009, AJ, 137, 4377
Yong, D., Carney, B. W., & Teixeira de Almeida, M. L. 2005, AJ, 130, 597
Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. 1997, ACM Trans. Math. Softw., 23, 550