Methods for the spectroscopic inference of fundamental stellar parameters

January 22, 2014

Ian Czekala[1] et al.

iczekala@cfa.harvard.edu

1. Introduction

The fundamental properties of stars—effective photospheric temperature, surface gravity, and metallicity–are crucial for our understanding of pre-main sequence stellar evolution (D'Antona & Mazzitelli 1997; Baraffe et al. 2002). The exoplanet community is also intensely focused on deriving accurate stellar properties because the properties of newly discovered exoplanets must necessarily be quoted relative to those of their host stars (e.g. Torres et al. 2012; Buchhave et al. 2012; Santos et al. 2013) and uncertain star properties translate into uncertain planet properties (Kane 2014). Many techniques[1] exist for stellar characterization such as line indexes, machine learning, and spectral synthesis[2]. In order to derive accurate and robust stellar properties for our small sample (~ 20) of T Tauri stars (some binary), we have decided to forward model the observed spectra by drawing from a grid of synthetic spectra.

1.1. Pixel to pixel techniques using synthetic spectral libraries

Most forward modeling techniques compare pixels directly using some form of

$$\chi^2 = \sum_i \left[\frac{D_i - M_i}{\sigma_i} \right]^2 = \sum_i R_i^2 \tag{1}$$

---

[1]Harvard-Smithsonian Center for Astrophysics, 60 Garden Street MS 10, Cambridge, MA 02138

[1]If you are interested, I have also compiled a more verbose review of modern techniques described in the literature. However, for the sake of brevity I have omitted it from this document.

[2]Given infinite computational power, we would prefer to fit *all* of the parameters that completely describe a synthetic spectrum: atmospheric structure, individual elemental abundances, and atomic constants describing the opacity contribution of a line. Approximations to the synthesis and efficient MCMC sampling parallelized on a cluster might soon put this approach within reach.

Symbol	Description
i	index specifying a pixel
λ_i	wavelength corresponding to a given pixel i
$\vec{\theta}_{\star,\mathrm{grid}}$	fundamental stellar parameters, $T_{\mathrm{eff}}, \log(g), [\mathrm{Fe/H}], [\alpha/\mathrm{Fe}]$
	that parameterize a synthetic spectrum from the grid
$\vec{\theta}_{\star,\mathrm{post}}$	stellar parameters $v \sin i$, v_z, A_V, and R^2/d^2 that
	are applied during "post processing" of the synthetic spectrum
$\vec{\theta}_{\star}$	$\{\vec{\theta}_{\star,\mathrm{grid}}, \vec{\theta}_{\star,\mathrm{post}}\}$
$c_0, c_1, \ldots$	Chebyshev polynomial coefficients for residual flux calibration
$\vec{\theta}_{\mathrm{N}}$	the set of nuisance parameters composed of $\{c_0, c_1, \ldots, c_N\}$
$k(i\vert\vec{\theta}_{\mathrm{N}})$	polynomial function that multiplies a synthetic spectrum
$\vec{\theta}$	the parameters $\{\vec{\theta}_{\star,\mathrm{grid}}, \vec{\theta}_{\star,\mathrm{post}}, \vec{\theta}_{\mathrm{N}}\}$ that completely describe a model spectrum
D_i	data flux for a given pixel, $D(\lambda_i)$
$\boldsymbol{D}$	data vector comprised of all D_i, $i = \{0, \ldots, N\}$
M_i	model flux for a given pixel, $M(\lambda_i\vert\vec{\theta})$
$\boldsymbol{M}$	model vector comprised of all M_i, $i = \{0, \ldots, N\}$
σ_i	Poisson noise for a given pixel i
R_i	residuals $(D_i - M_i)/\sigma_i$

Table 1: Nomenclature used in this document

where the variables are defined in Table 1. A serious problem with this approach is that synthetic spectra are not perfect: systematic errors invalidate the χ^2 assumption and bias parameter estimates and their uncertainties.

Current methods either circumvent this problem by using high quality synthetic grids that were painstakingly tuned over a narrow wavelength range (Buchhave et al. 2012; Schoenrich & Bergemann 2013, $\lesssim 500$ Å), or they use less-perfect grids with full optical coverage and sigma clip the bad regions of the spectrum (Koleva et al. 2009; Mann et al. 2013). Masking (either implicitly or explicitly, respectively) is troublesome because, in the words of Mann et al. (2013), "regions with [only] modest agreement between the real and synthetic spectra [that might be masked] may contain more temperature information than regions with slightly better matches, and we have no a priori information about what spectral regions are the most temperature sensitive." Masking could very well destroy stellar information in the pursuit of a better fit.
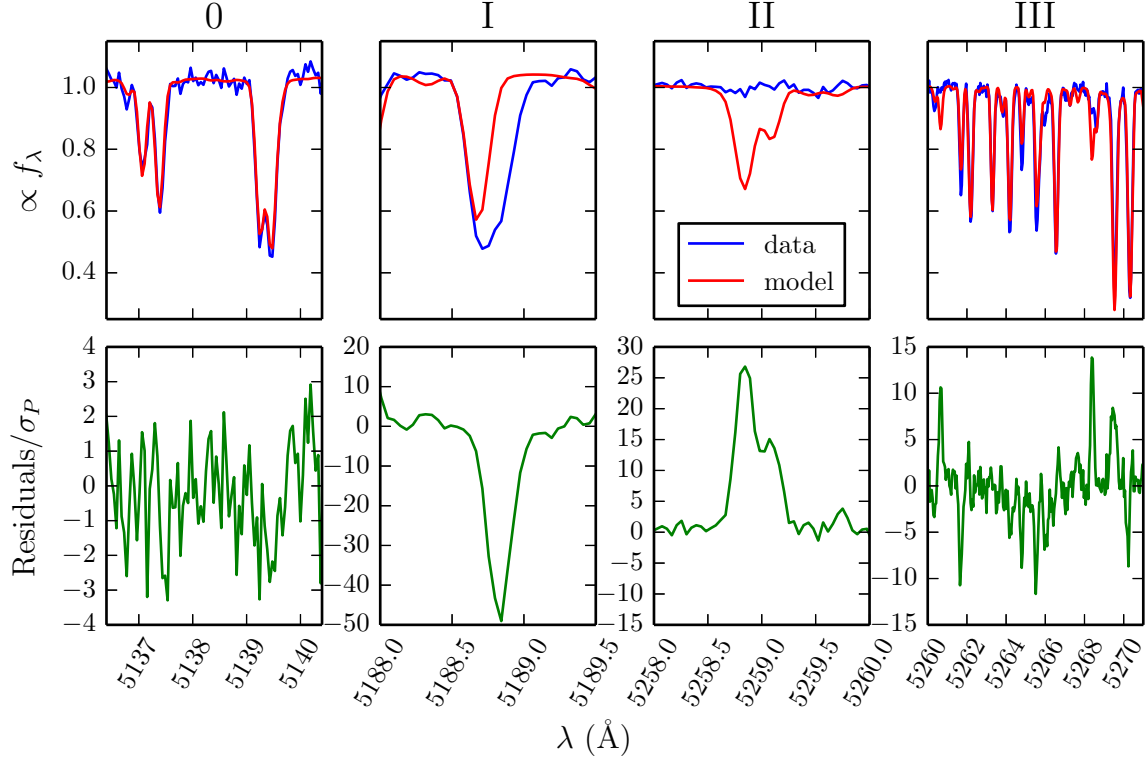
Fig. 1.— The four classes of line behavior. **Class 0**: the model behaves as expected. **Class I**: a spectral line is present in the data but missing from the model (in this case, the line is also blended). **Class II**: a spectral line exists in the model where there is only continuum in the data. **Class III**: lines exist in both the data and model, but some are too strong while others are too weak. The lower panels show the residual spectra in units of the Poisson uncertainty. The data is a spectrum of WASP-14, an F star. The model is from the PHOENIX grid(Husser et al. 2013).

1.2. Classes of synthetic spectral line behavior

Illustrated in Figure 1 are three ways that synthetic spectra often diverge from reality. When a spectral line appears in the data but not in the model, due to interstellar or telluric absorption or a missing opacity source, we call this a **Class I** error. When a spectral line appears in the model but not in the data, presumably due to a combination of incorrect atomic constants or atmospheric structure, we label it a **Class II** error. Both of these types of errors are present over a wide range of probable stellar parameters (1000 K in $T_{\rm eff}$,

and 1 dex in $\log(g)$, [Fe/H], and [α/Fe]) and there does not exist a $\vec{\theta}_{\star,\text{grid}}$ that could ever fit these lines, making these regions useless for determining stellar parameters. We know that for a normal AFGK star Class I residuals will always be negative and will not change significantly with $\vec{\theta}_{\star,\text{grid}}$, while Class II residuals will always be positive and their magnitude should decrease with decreasing metallicity. To help identify Class I and II errors, we could use the partial derivatives of an individual synthetic spectrum with respect to $\vec{\theta}_{\star,\text{grid}}$ and a prior on the range of parameters over which the residuals will be correlated to determine which lines show Class I or II behavior.

When some line strengths do not match at a $\vec{\theta}_{\star,\text{grid}}$ that properly fits most other lines, we term these **Class III** errors. These types of errors are pernicious because there does exist a $\vec{\theta}_{\star,\text{grid}}$ that would fit each line individually, but no one $\vec{\theta}_{\star,\text{grid}}$ can be chosen that fits all the lines. We must then decide which regions to trust and by how much, because Class III regions *still contain information about* $\vec{\theta}_{\star,\text{grid}}$. In fact, they might contain *the most* information about $\vec{\theta}_{\star,\text{grid}}$ since these delicate regions are the most sensitive to changes in $\vec{\theta}_{\star,\text{grid}}$ and are consequently whey they are the hardest to model correctly. In other words, Class III regions might be where the spectrum gradient ($\sum_j \partial M_i / \partial \vec{\theta}_{\star\text{grid},j}$) is the largest.

2. In the limit of a perfect stellar model

To generate a model spectrum $\boldsymbol{M}(\vec{\theta}_{\star,\text{grid}}, \vec{\theta}_{\star,\text{post}})$, we use $\vec{\theta}_{\star,\text{grid}}$ to linearly interpolate a synthetic spectrum from the grid, and then use $\vec{\theta}_{\star,\text{post}}$ to apply instrumental and rotational broadening in the Fourier domain, Doppler shift, correct for extinction, and then downsample to the exact pixels of the data spectrum using spline interpolation[3].

To account for imperfect flux calibration, we extend the approach of Eisenstein et al. (2006) and multiply $\boldsymbol{M}(\vec{\theta}_{\star,\text{grid}}, \vec{\theta}_{\star,\text{post}})$ by $k(\vec{\theta}_{\text{N}})$, which is a Chebyshev polynomial described by a set of N Chebyshev coefficients. We can either sample in the coefficients or analytically marginalize over them. We favor this approach over the traditional technique of normalizing to the stellar continuum due to concerns about correctly placing the pseudo-continuum in M stars. From now on we refer to $\boldsymbol{M}(\vec{\theta}) = k(\vec{\theta}_{\text{N}})\boldsymbol{M}(\vec{\theta}_{\star,\text{grid}}, \vec{\theta}_{\star,\text{post}})$. The mathematical details are in § A.

Currently we are sampling this χ^2 posterior with the MCMC ensemble sampler (Goodman & Weare 2010) implemented in `Python` by Foreman-Mackey et al. (2012) as `emcee`. In

---

[3]Because we are dealing with flux-calibrated data and synthetic spectra, we *resample* to lower resolution. We do not *rebin* because we are dealing with flux-density, not photon counts

this section, we described our approach to sample the posterior assuming perfect synthetic spectra. However, if one does a straight up χ^2 fit to our spectra, one can get posteriors that have 1-sigma contours of ± 1 K and ± 0.01 dex in gravity and metallicity. This would be as precise as stellar parameters obtained with astroseismology! Clearly there is something going wrong here.

3.   Beyond a perfect stellar model

3.1.   Additional sources of noise

- CCD readnoise and imperfect 1D spectrum extraction over night sky lines will almost definitely increase the noise in a pixel beyond theoretical Poisson limits. Normally this information will be encapsulated in a "sigma spectrum" output by IRAF, but unfortunately we do not have access to this for our current dataset.

- The stellar continuum can flicker/jitter (as shown by Kepler), which would produce a small departure from a perfect continuum. This might explain why even in the Class 0 region of our spectrum (Figure 1, left panel) the residuals are slightly larger than 1σ on average. For our T Tauri stars this effect may be more pronounced.

- When we interpolate a synthetic spectrum from the library to a specific $\vec{\theta}_{\star,\mathrm{grid}}$, we will introduce interpolation errors depending on how close we are to a grid point in the library. Husser (2012) investigate the possible range of interpolation errors at specific $\vec{\theta}_{\star,\mathrm{grid}}$ by dropping out a spectrum from the grid, interpolating over it, and then comparing the true spectrum to the interpolated spectra. For hotter stars with well defined continua, we find the maximum error is $\lesssim 5\%$, while for M stars the error could be as large as $\sim 20\%$ for some spectral features. By doing this interpolation test across the grid, we could determine an "interpolation error spectrum" that would encapsulate the maximum error possible for an interpolated spectrum, weighted by its distance from the nearest grid points. If a typical spectrum has signal/noise of ~ 30 per pixel (or noise/signal ratio of $1/30 \approx 3\%$), an interpolation noise contribution of $\sim 5\%$ the level of the signal level would be significant.

However, we have tried arbitrarily increasing the Poisson errors by a factor of 2 or 3 and it didn't significantly change the results, suggesting that something more drastic is needed beyond simply adding in more noise.

3.2. Gallery of likelihood functions

A problem with the χ^2 likelihood function is that the Gaussian tails rapidly decline, making outliers extremely improbable. A common approach for "robust regression" is to use a likelihood function with fatter, more forgiving, tails. Although not rigorously motivated, this is can be a quick thing to try. We explored several likelihood functions including a combination of Gaussian with exponential wings, a Lorentzian (Cauchy) function, and a function described by Sivia & Skilling (2006), which treats the noise estimate for each data point suspiciously. In Sivia's model, the true noise σ can be no smaller than some original estimate σ_0, but it may be much larger. By putting a prior on σ,

$$p(\sigma_i|\sigma_{0i}) = \left\{ \begin{array}{cc} \sigma_{0i}/\sigma_i^2 & \sigma_i \leq \sigma_{0i} \\ 0 & \text{else} \end{array} \right. \tag{2}$$

and marginalizing over the χ^2 likelihood, the new likelihood function becomes

$$p(D_i|M_i, \sigma_{0i}) = \frac{1}{\sigma_{0i}\sqrt{2\pi}} \left[\frac{1 - e^{-R_i^2/2}}{R_i^2} \right] \tag{3}$$

where $R_i = (D_i - M_i)/\sigma_{0i}$. This function has even fatter tails than the exponential model and seems to give results less biased by outliers, however it is not truly satisfying. Which lines contributed most to the posterior? What about the wings of class I and II errors? These individual pixels might still be given a weight of σ_{0i} because they are not nearly as bad as the line centers.

Other functions we might try in the future include a Student-t distribution, a linear combination of Gaussians, or the χ_Q^2 function described in Hogg et al. (2010). While "softer" likelihood functions do help somewhat, we would like to move beyond tweaking the fitting function towards an approach that properly incorporates our knowledge of the systematic errors of the synthetic grids. In the remaining two sections of this document we outline potential ideas that we might try next, but have not yet implemented in code.

4. Bad data mixture model

The bad data model (Press 1997; Hogg et al. 2010) is a powerful framework for forward modeling data, which allows the modeler to explicitly account for outliers using a generative model. In the simplest mixture model, each pixel has a binary flag q_i which specifies whether that data point is good or bad. Then, for the entire data set we have an ensemble

of $\{q_i\}_{i=1}^N$. If this ensemble of $\{q_i\}_{i=1}^N$ is chosen appropriately, and the good pixels are flagged as good and the bad pixels are flagged as bad, the posterior will be maximized. We adopt P_B for the *a priori* probability that a given pixel will be bad, Y_b for the mean flux of the bad pixels, and V_b for their variance. Written out explicitly, this is

$$\mathcal{L} = p(\boldsymbol{D}|\vec{\theta}, \{q_i\}_{i=1}^N, Y_b, V_b) = \prod_{i=1}^N \left[\frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(D_i - M_i)^2}{2\sigma_i^2} \right] \right]^{q_i} \times$$

$$\left[\frac{1}{\sqrt{2\pi(V_b + \sigma_i^2)}} \exp\left[-\frac{(D_i - Y_b)^2}{2(V_b + \sigma_i^2)} \right] \right]^{1-q_i} \tag{4}$$

If we put a Bernoulli prior on the q_i

$$p(q_i|P_b) = [1 - P_b]^{q_i} P_b^{[1-q_i]} \tag{5}$$

the probability of any given ensemble of $\{q_i\}_{i=1}^N$ is

$$p(\{q_i\}_{i=1}^N|P_b) = \prod_{i=1}^N [1 - P_b]^{q_i} P_b^{[1-q_i]} \tag{6}$$

We can analytically marginalize over the settings for a single flag by a discrete sum over the two states of $q_i = 1$ and $q_i = 0$.

$$p(\vec{\theta}, P_b, Y_b, V_b|D_i) = \int p(\vec{\theta}, q_i, P_b, Y_b, V_b|D_i) \mathrm{d}q_i$$

$$= p(\vec{\theta}, q_i = 1, P_b, Y_b, V_b|D_i) + p(\vec{\theta}, q_i = 0, P_b, Y_b, V_b|D_i)$$

$$= \left[p_{\mathrm{good}}(D_i|\vec{\theta}) \, p(q_i = 1|P_b) + p_{\mathrm{bad}}(D_i|Y_b, V_b) \, p(q_i = 0|P_b) \right] \times p(\vec{\theta}, P_b, Y_b, V_b) \tag{7}$$

$$p(\vec{\theta}, P_b, Y_b, V_b|\boldsymbol{D}) \propto \prod_{i=1}^N \left[[1 - P_b] p_{\mathrm{good}}(D_i|\vec{\theta}) + P_b p_{\mathrm{bad}}(D_i|Y_b, V_b) \right] \times p(\vec{\theta}, P_b, Y_b, V_b)$$

$$\boxed{= \prod_{i=1}^N \left\{ \frac{1 - P_b}{\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{(D_i - M_i)^2}{2\sigma_i^2} \right] + \frac{P_b}{\sqrt{2\pi(V_b + \sigma_i^2)}} \exp\left[-\frac{(D_i - Y_b)^2}{2(V_b + \sigma_i^2)} \right] \right\} \times p(\vec{\theta}, P_b, Y_b, V_b)}$$

$$\tag{8}$$

In its simplest form, the bad data model results in a likelihood function that is a linear combination of Gaussians. Although this functional form might be similar to some

of the likelihood functions in § 3.2, the benefit of this bad data approach over a gallery of likelihood functions is that it is motivated by the characteristics of the bad data (P_B, Y_b, and V_b). If we wish to add additional information about the characteristics of the bad data, we can do so.

4.1. Mixture of multiple components

When we believe that a sample population might be composed of several populations with different behaviours (Class 0, I, II, and III), we can expand the bad data model to a multiple mixture model. Following Gelman et al. (2004), we develop a mixture model with $M = 4$ components, one for each class of error. Our likelihood function is then

$$p(\boldsymbol{D}, \vec{Q}|\vec{\theta}, J) = \prod_i^N \prod_{m=0}^M \left[J_m \mathcal{L}_m(D_i|\vec{\theta}_m) \right]^{q_{im}} \tag{9}$$

where $\vec{Q} = \{q_{im}\}_{i=1,m=0}^{N,M}$ is a two dimensional array ($N \times M$) of all the individual q_{im} flags, and J_m is a weight that represents the total fraction of each component in the mixture[4], and $\mathcal{L}_m$ is the likelihood function for mth mixture component. At this point, $\vec{\theta}$ now contains all of the parameters that might be used by any of the likelihood functions. $\vec{\theta}_m$ is used to denote only those parameters that will be used for $\mathcal{L}_m$. Given what we know about these synthetic spectral errors, we could specify the following likelihood functions

$$\begin{aligned}
\mathcal{L}_0 &= p(D_i, q_{i0} = 1) &\rightarrow& \quad \text{Class 0} & \chi^2 \\
\mathcal{L}_1 &= p(D_i, q_{i1} = 1) &\rightarrow& \quad \text{Class 1} & \text{flat lnprob} \\
\mathcal{L}_2 &= p(D_i, q_{i2} = 1) &\rightarrow& \quad \text{Class 2} & \text{flat lnprob} \\
\mathcal{L}_3 &= p(D_i, q_{i3} = 1) &\rightarrow& \quad \text{Class 3} & \text{soft } \chi^2
\end{aligned} \tag{10}$$

This would treat Class I and II errors as if they had been masked from the dataset but allow Class III to have a slight influence on the posterior since these regions still contain information about $\vec{\theta}_\star$. The softening term of the χ^2 could even be sampled for, much like the V_b of the original bad data model.

4.2. Spatial correlations of pixels

Previously we assumed that each data point is independent, but we know that the spectrum is spatially correlated across several pixels with a correlation length set by the

---

[4]The weights are normalized $\sum_m J_m = 1$

width of the convolution of the stellar rotation kernel ($v \sin i$) and the instrumental resolution kernel, which is also the typical width of a spectral line[5]. This correlation length will have some dependence on the spectral type of star. For example, F and G stars will likely have singular, well-defined lines while M stars might have a large molecular feature covering a wide swath of the spectrum. Neighboring pixels are actually correlated in two coordinates: the q_{im} flags and the residuals R_i.

4.2.1.   Correlation of the q_i flags

The pixels at the center of a bad line will give a strong residual response, which will likely be flagged as bad. However, the pixels in the wings of the spectral line might not be sufficiently egregious to be properly flagged. To remedy this situation, we attempt to develop a bad data model which includes spatial (wavelength) correlations.

In the manner of Gaussian random processes (GRP), we develop a covariance function

$$\mathrm{cov}(q_i, q_j) = \rho(|i - j|) \tag{11}$$

$$\Sigma_{ij} = \sigma^2 \rho(|i - j|) \tag{12}$$

that describes the correlation between the state of the bad data flags. The simplest thing to do would be to assume a common correlation length in the spectrum. If we say that the typical feature will have three contiguous pixels[6] wrong, then the correlation function would be

$$\mathrm{cov}(q_i, q_j) = \begin{cases} 1 - \frac{|i-j|}{3} & |i - j| < 3 \\ 0 & \text{else} \end{cases} \tag{13}$$

---

[5]Some very strong lines (such as the Balmer lines or Mg b triplet) will have wings with a larger correlation length.

[6]Ideally the covariance function should be a function over λ_i in velocity space, since all spectral features should have a similar velocity width. For the purpose of keeping the math simple for the moment, we use pixel index and assume the pixels are uniformly spaced.

and the matrix would be

$$
\begin{bmatrix}
1 & 2/3 & 1/3 & 0 & \cdots & 0 \\
2/3 & 1 & 2/3 & \ddots & \ddots & \vdots \\
1/3 & 2/3 & \ddots & \ddots & \ddots & 0 \\
0 & \ddots & \ddots & \ddots & 2/3 & 1/3 \\
\vdots & \ddots & \ddots & 2/3 & 1 & 2/3 \\
0 & \cdots & 0 & 1/3 & 2/3 & 1
\end{bmatrix}
\tag{14}
$$

Because our correlation function has compact support, we now have a sparse covariance matrix. This can potentially speed up a matrix computation that would otherwise be very slow.

4.2.2. Correlation of adjacent residuals

Not only will the flags q_i be spatially correlated but so will the values of the residuals R_i. For OBAFGK stars, the residuals are likely to be shaped like Gaussians. If i specifies the pixel at the line center of a large residual feature, then the neighboring $i \pm 1, 2, \ldots$ pixels will have also have similar residuals but of declining magnitude $|R_{i\pm1,2,\ldots}| \lesssim |R_i|$. For M stars, there might be a wider correlation length and the shape of the lines might not be well-described by a Gaussian.

I am new to the field of GRP, but it seems like it might be possible to come up with a covariance function that would describe the predicted heights of the residuals and regress against them. The covariance function between R_i could be learned from the spectrum itself. Then, the likelihood functions for each mixture component in the model would be custom tailored for the type of error that it expected to encounter. For example, Class 0 residuals would be flat while Class I, II, and III residuals would have some sort of Gaussian shape. Class I residuals would be negative while Class II residuals would be positive. I plan to further investigate covariance functions.

4.2.3. Further ideas

- Gelman et al. (2004, ch. 18) list several references further describing inference for spatially-correlated data and I plan to work my way through these.

- Naively, it seems to me like there might be a way to use FFT methods to help speed up the correlation calculation, similar to Gaussian random fields.

- My preliminary experiments with GRP reveal that there is an art in choosing the functional form of $\text{cov}(i, j)$ and setting the hyperparameters. Perhaps these could also be parameters that we sample in. I am working through Rasmussen & Williams (2005) and Gelman et al. (2013, Ch 21.) to learn more.

4.3. Hyperparameters

Ma & Berndsen (2013) discuss a technique to deal with systematic errors when combining different data sets for the combined inference of a single problem. They use a hyperparameter approach that can artificially inflate the variance of each data set to find a common posterior and allow for correlations between data sets. We could set up a similar hyperparameter approach that can tune the likelihood contribution for certain lines, treating each line as they do different datasets. If all of the lines were in harmonious agreement, this hyperparameter would undergo "shrinkage" (Kruschke 2010) and tighten the posterior on $\vec{\theta}_\star$. If the line fits were incompatible, this hyperparameter would increase while also broadening our posterior on $\vec{\theta}_\star$.

4.4. Sampling ideas for the mixture model with spatial correlations

If we use multidimensional Gaussian likelihood functions and correlation functions with compact support, it might still be possible to analytically marginalize over the multiple mixture components. If we cannot marginalize, then we need alternate sampling methods to explore the parameter space in a computationally efficient manner. A stellar model with flags for each pixel would have $\vec{\theta}_\star + (N \times M)$ parameters, where for a typical high resolution optical spectrum $N > 100{,}000$.

An alternative way to approach this problem is to do Gibbs sampling where we propose a jump in a single q_{im} at a time. Such a sample requires none of the previous heavy lifting (spectral grid interpolation, rotational broadening, or pixel interpolation), and so even though there are $\sim 100{,}000$ flags to sample, this might be computationally feasible.

In order to "trim the tree," Gelman et al. (2004) suggest to start with a crude estimate (such as k-means clustering) to tentatively identify which observations have been drawn from which mixture components. They also suggest that the expectation-maximize (EM)

algorithm could also be used to sample this posterior, although this is an algorithm that I need to research further.

5.   Injection of fake lines

As an alternative approach to modeling the regions of the spectrum suffering from systematics, we could attempt to patch the models by injecting fake lines into the spectrum to compensate the errors. Each line could be a Gaussian parameterized by λ_0, strength, and FWHM. The line strength distribution could be described by a hyper prior, and the FWHM described by a hyper parameter, since all lines should have pretty much the same FWHM. We could penalize the model fits by having a prior on the number of bad lines that can be created. Given a bad line and the linelist that produced the synthetic models, it may be possible to identify other regions of the spectrum that might be bad. For example, if a vanadium line is bad, then we could check to see if the other vanadium lines might also be bad.

Class III errors might be difficult to address in this paradigm since we would be fitting the difference of two lines, but if they are both Gaussians of the same FWHM, then it should still work. In M star spectra, rather than adding a Gaussian, we might need to add a larger continuum feature, which might be difficult to parameterize.

5.1.   Piecewise metallicity model

In theory, it is possible to fit the profiles of only certain elements to determine the abundances of these elements. By masking all non-alpha elements, Husser (2012) use PHOENIX spectra (Husser et al. 2013) to derive alpha abundances of stars. Rather than masking, we could assemble a piecewise spectrum where regions of the spectrum might be drawn from a different combination of $\vec{\theta}_{\star,\mathrm{grid}}$, say with lower metallicity for some lines in order to account for a bad line strength. These regions would have to be normalized properly so that the composite spectrum is sufficiently smooth. Coupled with a linelist, one could also tweak individual elemental abundances.

A.   A technique to account for imperfect flux calibration

The first four Chebyshev polynomials are

$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_2(x) = 2x^2 - 1$$
$$T_3(x) = 4x^3 - 3x$$

In our implementation, we map the full pixel range of a specific order $i \in [i_{\min}, i_{\max}]$ to $x \in [-1, 1]$. By mapping to pixel number rather than wavelength, we are insensitive to Doppler shifts of the model. If the spectrum has multiple orders, like echelle spectra, we would have a set of Chebyshev coefficients $\vec{\theta}_N = \{c_0, c_1, c_2, \ldots, c_N\}$ for each order.

$$k(i|\vec{\theta}_N) = c_0 T_0(i) \left[1 + c_1 T_1(i) + \ldots + c_N T_N(i) \right] \tag{A1}$$

$$k(i|\vec{\theta}_N) M_i = \left[1 + \sum_{j=1}^{N} c_j T_j(i) \right] (c_0 M_i) \tag{A2}$$

$$\tag{A3}$$

This functional form of k allows c_0 to scale by large amounts while the higher order terms are perturbations on top of the *scaled* model ($c_0 M_i$). This formulation permits large rescalings up or down without biasing the higher order terms. If c_0 is expected to be small, we can use a simpler implementation described in §A.2. Then for the full spectrum

$$\ln\left[p(\vec{\theta}|\boldsymbol{D})\right] \propto -\frac{1}{2} \sum_i \left[\frac{D_i - \left[1 + \sum_{j=1}^{N} c_j T_j(i)\right] c_0 M_i}{\sigma_i} \right]^2 \tag{A4}$$

To simplify the following discussion, we consider a single pixel i and then generalize this to a sum over all pixels later. We expand out the square in Equation A4

$$\ln\left[p(\vec{\theta}|D_i)\right] \propto -\frac{1}{2\sigma_i^2} \left[D_i^2 - 2D_i c_0 M_i \left[1 + \sum_{j=1}^{N} c_j T_j(i) \right] + c_0^2 M_i^2 \left[1 + 2\sum_{j=1}^{N} c_j T_j(i) + \sum_{j=1}^{N} \sum_{k=1}^{N} c_j T_j(i) c_k T_k(i) \right] \right] \tag{A5}$$

To simplify the notation, we can rewrite the higher order Chebyshev coefficients and polynomials as column vectors $\vec{c}$ and $\vec{T}(i)$, respectively

$$\vec{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} \qquad\qquad \vec{T}(i) = \begin{bmatrix} T_1(i) \\ T_2(i) \\ \vdots \\ T_N(i) \end{bmatrix} \tag{A6}$$

then

$$k(i|\vec{\theta}_{\mathrm{N}}) = c_0 \left[1 + \sum_{j=0}^{N} c_j T_j(i) \right] = c_0 \left[1 + \vec{T}^{\mathsf{T}} \cdot \vec{c} \right] \tag{A7}$$

If we let

$$\boldsymbol{W}(i) = \vec{T} \cdot \vec{T}^{\mathsf{T}} = \begin{bmatrix} T_1 T_1 & T_1 T_2 & \dots & T_1 T_N \\ T_2 T_1 & T_2 T_2 & \dots & T_2 T_N \\ \vdots & \vdots & \ddots & \vdots \\ T_N T_1 & T_N T_2 & \dots & T_N T_N \end{bmatrix} \tag{A8}$$

then we have

$$\sum_{j=1}^{N} \sum_{k=1}^{N} c_j T_j(i) c_k T_k(i) = \vec{c}^{\mathsf{T}} \cdot \boldsymbol{W} \cdot \vec{c} \tag{A9}$$

We can rewrite Equation A5 quadratically in $\vec{c}$ as

$$\ln\left[p(\vec{\theta}|D_i) \right] \propto -\frac{1}{2} \frac{c_0^2 M_i^2}{\sigma_i^2} \vec{c}^{\mathsf{T}} \boldsymbol{W} \vec{c} + \left(\frac{-c_0^2 M_i^2 + c_0 D_i M_i}{\sigma_i^2} \right) \vec{T}^{\mathsf{T}} \vec{c} - \left(\frac{c_0^2 M_i^2 - 2 c_0 D_i M_i + D_i^2}{2 \sigma_i^2} \right) \tag{A10}$$

Because matrix multiplication is associative, we can sum $\boldsymbol{W}(i)$ and $\vec{T}^{\mathsf{T}}(i)$ across all pixels and define

$$\boldsymbol{A} = c_0^2 \sum_i \frac{M_i^2}{\sigma_i^2} \boldsymbol{W}(i) \tag{A11}$$

$$\vec{B} = \sum_i \frac{-M_i^2 c_0^2 + D_i M_i c_0}{\sigma_i^2} \vec{T}(i) \tag{A12}$$

$$g = -\frac{1}{2} \sum_i \frac{c_0^2 M_i^2 - 2 c_0 D_i M_i + D_i^2}{\sigma_i^2} \tag{A13}$$

Where $\boldsymbol{A}$, $\vec{B}$, and g are each a function of $\vec{\theta}_{\star}$, $\vec{\theta}_{\mathrm{N}}$, and $\boldsymbol{D}$. Now we can rewrite Equation A4 as

$$p(\vec{\theta}|\boldsymbol{D}) = p(\vec{\theta}_{\star}, \vec{\theta}_{\mathrm{N}}|\boldsymbol{D}) \propto \exp\left(-\frac{1}{2} \vec{c}^{\mathsf{T}} \boldsymbol{A} \vec{c} + \vec{B}^{\mathsf{T}} \vec{c} + g \right) \tag{A14}$$

Generally, we are most interested in the stellar parameters after they have been marginalized over the nuisance parameters. Because this probability function is a multi-dimensional Gaussian in the higher order Chebyshev coefficients $\vec{c} = \{c_1, c_2, \ldots, c_N\}$, we can do this marginalization analytically (Stoof et al. 2009), and then sample from the marginalized distribution.

$$p(\vec{\theta}_\star, c_0 | \boldsymbol{D}) = \int p(\vec{\theta}_\star, c_0, \vec{c} \, | \boldsymbol{D}) \, \mathrm{d}\vec{c} \tag{A15}$$

$$p(\vec{\theta}_\star, c_0 | \boldsymbol{D}) \propto \int \exp\left(-\frac{1}{2}\vec{c}^{\mathsf{T}} \boldsymbol{A} \vec{c} + \vec{B}^{\mathsf{T}} \vec{c} + g\right) \mathrm{d}\vec{c} \tag{A16}$$

$$p(\vec{\theta}_\star, c_0 | \boldsymbol{D}) \propto \sqrt{\frac{(2\pi)^N}{\det|\boldsymbol{A}|}} \exp\left(\frac{1}{2}\vec{B}^{\mathsf{T}} \boldsymbol{A}^{-1} \vec{B} + g\right) \tag{A17}$$

By doing this marginalization we drastically reduce the dimensionality of our parameter space, especially if our echelle spectrum has multiple orders.

A.1.   Including Gaussian priors on the nuisance parameters

Using this polynomial formalism, it is possible to incorporate prior knowledge about the degree of necessary "re-fluxing" through priors on the nuisance parameters, determined by observing several spectrophotometric standard stars. Because c_0 is a scale factor, it should be equally probable to scale up by a factor of two as it is to scale down by a factor of two. This means we should be using a log-normal prior on c_0

$$p(c_0) = \frac{1}{\sqrt{2\pi}\sigma_{c_0} c_0} \exp\left(-\frac{(\ln c_0)^2}{2\sigma_{c_0}^2}\right) \tag{A18}$$

centered on $c_0 = 1$. Because the higher order polynomial terms are perturbations on the scaled model ($c_0 M_i$), we assume Gaussian priors centered about 0

$$p(c_n) = \frac{1}{\sqrt{2\pi}\sigma_{c_n}} \exp\left(-\frac{c_n^2}{2\sigma_{c_n}^2}\right) \tag{A19}$$

which we can express in vector form as

$$p(\vec{c}) \propto \exp\left(-\frac{1}{2}(\vec{c} - \vec{\mu})^{\mathsf{T}} \boldsymbol{D}(\vec{c} - \vec{\mu})\right) \tag{A20}$$

where

$$
\boldsymbol{D} = \begin{bmatrix}
\sigma_1^{-2} & 0 & \ldots & 0 \\
0 & \sigma_2^{-2} & \ldots & 0 \\
\vdots & \vdots & \ddots & 0 \\
0 & 0 & \ldots & \sigma_N^{-2}
\end{bmatrix}
\tag{A21}
$$

and σ_j represents the width of the Gaussian prior on the j-th Chebyshev coefficient. $\vec{\mu}$ is the mean of the Gaussian prior, in nearly all cases we expect to set this to $\vec{\mu} = \{0, 0, \ldots, 0\}$, unless we have prior knowledge that the flux-calibration is systematically warped in one direction. Now, we can write the full posterior with priors as

$$
p(\vec{\theta}_\star, \vec{\theta}_N | \boldsymbol{D}) \propto \exp\left(-\frac{1}{2}\vec{c}^\mathsf{T} \boldsymbol{A}\vec{c} + \vec{B}^\mathsf{T}(\vec{\theta}_\star)\vec{c} + g(\vec{\theta}_\star)\right) \exp\left(-\frac{1}{2}(\vec{c} - \vec{\mu})^\mathsf{T} \boldsymbol{D}(\vec{c} - \vec{\mu})\right) p(\vec{\theta}_\star)p(c_0)
\tag{A22}
$$

We can expand and rearrange the argument of Equation A20 to a similar form that is quadratic in $\vec{c}$ rewrite

$$
\boldsymbol{A}' = \boldsymbol{A} + \boldsymbol{D}
\tag{A23}
$$

$$
\vec{B}' = \vec{B} + (\boldsymbol{D}\vec{\mu})^\mathsf{T}
\tag{A24}
$$

$$
g' = g - \frac{1}{2}\vec{\mu}^\mathsf{T} \boldsymbol{D}\vec{\mu}
\tag{A25}
$$

in the case that $\vec{\mu} = \{0, 0, \ldots, 0\}$, there is only a non-zero correction to $\boldsymbol{A}$. The full posterior probability distribution is then

$$
\boxed{p(\vec{\theta}_\star, \vec{\theta}_N | \boldsymbol{D}) = p(\vec{\theta}_\star, c_0, \vec{c}) \propto \exp\left(-\frac{1}{2}\vec{c}^\mathsf{T} \boldsymbol{A}'\vec{c} + \vec{B}'^\mathsf{T}\vec{c} + g'\right) p(\vec{\theta}_\star)p(c_0)p(\vec{c})}
\tag{A26}
$$

and the margin form is

$$
\boxed{p(\vec{\theta}_\star, c_0 | \boldsymbol{D}) \propto \sqrt{\frac{(2\pi)^N}{\det|\boldsymbol{A}'|}} \exp\left(\frac{1}{2}\vec{B}'^\mathsf{T} \boldsymbol{A}'^{-1}\vec{B}' + g'\right) p(\vec{\theta}_\star)p(c_0)}
\tag{A27}
$$

If one uses MCMC to sample the marginalized posterior, but desires a flux-corrected spectrum for the purposes of inspecting the residuals, it is still possible to do. We can draw a random sample $\vec{\theta}_\star$ from the MCMC chain and then resample the unmarginalized posterior (Eqn A26) conditional on $p(\vec{\theta}_N | \vec{\theta}_\star, \boldsymbol{D})$. Now, the combination of $\vec{\theta}_\star$ and $\vec{\theta}_N$ completely describes a flux-corrected spectrum.

A.2. Simplification for small c_0 perturbations

If the correction in c_0 is small, we can use an alternate formulation of k and put Gaussian priors on all nuisance parameters. This allows analytic marginalization over all nuisance parameters (including c_0), and reduces the dimensionality of the posterior to only the stellar parameters, $\vec{\theta}_\star$.

$$k(\lambda|\vec{\theta}_N) = c_0 T_0(\lambda) + c_1 T_1(\lambda) + \ldots + c_N T_N(\lambda) \tag{A28}$$

$$k(\lambda|\vec{\theta}_N) = \sum_{i=0}^{N} c_i T_i(\lambda) \tag{A29}$$

$$\ln\left[p(\vec{\theta}|\lambda_i)\right] \propto -\frac{1}{2\sigma^2}\left[\boldsymbol{D}^2 - 2\boldsymbol{DM}\sum_n c_n T_n + \boldsymbol{M}^2\sum_n\sum_m c_n T_n c_m T_m\right] \tag{A30}$$

Now the column vectors $\vec{c}$ and $\vec{T}(\lambda)$ include c_0 as well

$$\vec{c} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} \qquad\qquad \vec{T}(\lambda) = \begin{bmatrix} T_0(\lambda) \\ T_1(\lambda) \\ \vdots \\ T_N(\lambda) \end{bmatrix} \tag{A31}$$

$$k(\lambda|\vec{\theta}_N) = \sum_{i=0}^{N} c_i T_i = \vec{T}^{\mathsf{T}} \cdot \vec{c} \tag{A32}$$

$$\boldsymbol{W}(\lambda) = \vec{T} \cdot \vec{T}^{\mathsf{T}} = \begin{bmatrix} T_0 T_0 & T_0 T_1 & \ldots & T_0 T_N \\ T_1 T_0 & T_1 T_1 & \ldots & T_1 T_N \\ \vdots & \vdots & \ddots & \vdots \\ T_N T_0 & T_N T_1 & \ldots & T_N T_N \end{bmatrix} \tag{A33}$$

$$\sum_n\sum_m c_n T_n c_m T_m = \vec{c}^{\mathsf{T}} \cdot \boldsymbol{W} \cdot \vec{c} \tag{A34}$$

$$\boldsymbol{A} = \sum_\lambda \frac{\boldsymbol{M}^2(\lambda)}{\sigma^2(\lambda)} \boldsymbol{W}(\lambda) \tag{A35}$$

$$\vec{B}^{\mathsf{T}} = \sum_\lambda \frac{\boldsymbol{D}(\lambda)\boldsymbol{M}(\lambda)}{\sigma^2(\lambda)} \vec{T}^{\mathsf{T}}(\lambda) \tag{A36}$$

$$g = -\frac{1}{2}\sum_\lambda \frac{\boldsymbol{D}(\lambda)}{\sigma^2(\lambda)} \tag{A37}$$

$$\tag{A38}$$

If we include Gaussian priors, then we have

$$\boldsymbol{A}' = \boldsymbol{A} + \boldsymbol{D} \tag{A39}$$

$$\vec{B}' = \vec{B} + (\boldsymbol{D}\vec{\mu})^{\mathsf{T}} \tag{A40}$$

$$g' = g - \frac{1}{2}\vec{\mu}^{\mathsf{T}}\boldsymbol{D}\vec{\mu} \tag{A41}$$

where now $\boldsymbol{D}$ and $\vec{\mu}$ include c_0 as well. The main difference from before is that now the prior on c_0 is now Gaussian instead of log-normal. This is acceptable as long as σ_{c_0} is small ($< XX\%$), otherwise this will bias the correction factor. This also means that we will usually have $\vec{\mu} = \{1, 0, \ldots, 0\}$. The full posterior probability function becomes

$$p(\vec{\theta}_{\star}, \vec{\theta}_{\mathrm{N}} | \boldsymbol{D}) = p(\vec{\theta}_{\star}, \vec{c}\,) \propto \exp\left(-\frac{1}{2}\vec{c}^{\mathsf{T}}\boldsymbol{A}'\vec{c} + \vec{B}'^{\mathsf{T}}\vec{c} + g'\right)p(\vec{\theta}_{\star})p(\vec{c}) \tag{A42}$$

and the marginal form is

$$p(\vec{\theta}_{\star} | \boldsymbol{D}) \propto \sqrt{\frac{(2\pi)^N}{\det|\boldsymbol{A}'|}}\,\exp\left(\frac{1}{2}\vec{B}'^{\mathsf{T}}\boldsymbol{A}'^{-1}\vec{B}' + g'\right)p(\vec{\theta}_{\star}) \tag{A43}$$

To gain insight, we can re-express this new version of k in the original form as

$$k(\lambda | \vec{\theta}_{\mathrm{N}}) = c_0 T_0 + c_1 T_1 + \ldots + c_N T_N \tag{A44}$$

where $c_1 = c_0 c_1' T_0$ and $c_N = c_0 c_N' T_0$. Thus, if there needs to be a large perturbation in c_0, then there must also be a large perturbation in c_n. The prior on c_1 is now a product of a log-normal prior on c_0 and a Gaussian prior on c_1', which is not analytically tractable. Therefore, in the case of large perturbations, it is better to use Equation and use a log-normal prior on c_0. When we use the original framework, we are saying that perturbations in c_0 are small enough that we can approximate the log-normal prior on c_0 as Gaussian and that the priors on c_n can also be correctly approximated by a Gaussian. From tests, the original framework gives acceptable accuracy (using $\sigma_{c_n} = 0.05$) if the corrections in c_0 are $\sigma_{c_0} \leq 0.35$, or the one-sigma deviation of c_0 is between $0.7 \leq c_0 \leq 1.42$.

REFERENCES

Baraffe, I., Chabrier, G., Allard, F., & Hauschildt, P. H. 2002, A&A, 382, 563, arXiv:astro-ph/0111385

Buchhave, L. A. et al. 2012, Nature, 486, 375

D'Antona, F., & Mazzitelli, I. 1997, Mem. Soc. Astron. Italiana, 68, 807

Eisenstein, D. J. et al. 2006, ApJS, 167, 40, arXiv:astro-ph/0606700

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2012, ArXiv e-prints, 1202.3665

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., & Rubin, D. 2013, Bayesian Data Analysis, Third Edition, Chapman & Hall/CRC Texts in Statistical Science (Taylor & Francis)

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. 2004, Bayesian data analysis (Chapman & Hall/CRC)

Goodman, J., & Weare, J. 2010, Communications in Applied Mathematics and Computational Science, 5, 65

Hogg, D. W., Bovy, J., & Lang, D. 2010, ArXiv e-prints, 1008.4686

Husser, T.-O. 2012, 3D-Spectroscopy of Dense Stellar Populations (Universitätsverlag Göttingen)

Husser, T.-O., Wende-von Berg, S., Dreizler, S., Homeier, D., Reiners, A., Barman, T., & Hauschildt, P. H. 2013, A&A, 553, A6, 1303.5632

Kane, S. R. 2014, ArXiv e-prints, 1401.3349

Koleva, M., Prugniel, P., Bouchard, A., & Wu, Y. 2009, A&A, 501, 1269, 0903.2979

Kruschke, J. 2010, Doing Bayesian Data Analysis: A Tutorial Introduction with R (Academic Press)

Ma, Y.-Z., & Berndsen, A. 2013, ArXiv e-prints, 1309.3271

Mann, A. W., Gaidos, E., & Ansdell, M. 2013, ApJ, 779, 188, 1311.0003

Press, W. H. 1997, in Unsolved Problems in Astrophysics, ed. J. N. Bahcall & J. P. Ostriker, 49–60, arXiv:astro-ph/9604126

Rasmussen, C. E., & Williams, C. K. I. 2005, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning) (The MIT Press)

Santos, N. C. et al. 2013, A&A, 556, A150, 1307.0354

Schoenrich, R., & Bergemann, M. 2013, ArXiv e-prints, 1311.5558

Sivia, D., & Skilling, J. 2006, Data analysis: a Bayesian tutorial (Oxford University Press, USA)

Stoof, H. T., Gubbels, K. B., & Dickerscheid, D. B. 2009, Ultracold quantum fields (Springer)

Torres, G., Fischer, D. A., Sozzetti, A., Buchhave, L. A., Winn, J. N., Holman, M. J., & Carter, J. A. 2012, ApJ, 757, 161, 1208.1268