

A method for inference of fundamental stellar parameters using high resolution spectra

January 15, 2014

Ian Czekala¹ et al.

iczekala@cfa.harvard.edu

ABSTRACT

Matching observed spectra of stars against a grid synthetic spectra is a popular approach to determine fundamental stellar parameters such as effective temperature, surface gravity, and metallicity. However, a fundamental problem with this approach is that the synthetic spectra contain systematic errors such as wrong or missing spectral lines. In the presence of such errors, a simple χ^2 fit between spectra will likely yield biased parameters and incorrect uncertainty estimates. This behaviour is analogous to the way a few strong outliers might severely bias the fit to a straight line. These spectral outliers are difficult to identify *a priori*; procedures such as sigma clipping might be prone to their own systematic effects and discard useful information. We outline an approach to probabilistically identify and account for spectral features that deviate from reality. Such an approach is most needed when operating at high spectral resolution ($R > 10,000$), since this is when systematic deviations begin to appear most striking, however our technique is still valid for lower resolution spectra as well. We also outline some approaches for spectral fits to brown dwarfs and planets, where instead of spectral lines there may be systematic deviations of a larger swath of the spectrum.

1. Introduction

The fundamental parameters of stars—mass, luminosity, effective photospheric temperature, surface gravity, and metallicity—are of supreme interest to astrophysicists for a wide variety of reasons. A salient example is that most newly discovered exoplanets must

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

necessarily their mass and radius measured relative to the properties of their host stars (e.g. Torres et al. 2012; Buchhave et al. 2012; Santos et al. 2013). The properties of stars are of course also important in their own right because they inform us about our ideas of stellar evolution (D’Antona & Mazzitelli 1997; Baraffe et al. 2002). Because stars are the fundamental units of the cosmos, techniques to infer the properties of stars date back to the dawn of astronomy. We briefly review some of the modern and popular techniques in the literature. Approaches can be broadly classified into techniques that compare stars by some broadly defined indices and techniques that attempt to forward-model an observed stellar spectrum by doing a pixel-to-pixel comparison.

1.1. Similarity techniques

Many of existing techniques were written to analyze low to medium resolution ($R < 10,000$) spectra. I am calling them similarity techniques because they compare characteristics of spectra, not actual pixels.

Line indexes By cataloguing the equivalent widths of chosen spectral lines into indexes, one can correlate the index with stellar properties and identify features that are sensitive temperature, surface gravity, or metallicity. Then, one matches the indecies of a new star against these relationships and derives a set of stellar parameters. A recent example using M dwarfs is Newton et al. (2014). This technique works, but there are many things that could be improved. First, the selection of which lines to measure is arduous and based upon expert knowledge of previously catalogued stars. This is both a disadvantage and an advantage. On one hand by choosing a subset of lines to fit, one may be unwittingly throwing away useful information. On the other hand, it allows the modeller to discard regions of the spectrum that do not follow a simple trend with stellar parameters. Second, while measuring equivalent widths is easy to do, this discards any useful information contained in the shape of the line profile.

Principal component analysis PCA reduces the dimensionality of the data space and searches for the most salient features of the spectra (Paletou et al. 2014). How does one properly estimate parameter uncertainties in such an approach? It seems like this approach exists because it is an easy thing to do.

Matrix inversions: MATISSE (Recio-Blanco et al. 2006). This method is similar to PCA in that it derives basis vectors that describe the fundamental parameters of the

star. In the PCA analysis, the principal components do not necessarily correspond to true physical quantities. The MATISSE algorithm determines weights that ensure the components are similar to the stellar parameters. They only test the algorithm against input synthetic stellar spectra to detect biases in the parameters, and against the Sun and Arcturus. They do acknowledge that there will be systematic deviations between the synthetic spectra and real spectra, but they do not account for the possible inflation of error estimates.

Given the inherent systematic uncertainties involved in each method, some groups elect to combine the inference from different algorithms, such as (SLOAN SSPP) (Lee et al. 2008). This approach includes machine learning along with many of the previously mentioned techniques.

1.2. Pixel to pixel techniques using spectral libraries

Many high quality synthetic spectral grids are now available. Nearly all synthetic stellar libraries are parameterized by the “fundamental” stellar parameters of effective temperature, surface gravity, metallicity, and alpha enhancement. We define this combination as

$$\vec{\theta}_{\star, \text{grid}} = \{T_{\text{eff}}, \log(g), [\text{Fe}/\text{H}], [\alpha/\text{Fe}]\} \quad (1)$$

Naively, it makes sense to compare an observed spectrum against spectra from the synthetic grid. Using some goodness of fit statistic one can identify the $\vec{\theta}_{\star, \text{grid}}$ that best reproduce the observed spectrum.

The pixel-to-pixel likelihood function will evaluate the residuals in some form of

$$\text{residual} = \frac{\text{data} - \text{model}}{\text{noise}} \quad (2)$$

where the best model is defined as one that minimizes the residuals in some sense. We will discuss other goodness of fit functions soon. Each of these quantities are functions of wavelength λ , so moving forward we will refer to them as the data spectrum $f_{\text{D}}(\lambda)$, the model spectrum $f_{\text{M}}(\lambda|\vec{\theta})$, the noise spectrum $\sigma(\lambda)$, and the residual spectrum $R(\lambda|\vec{\theta})$.

Most pixel-to-pixel techniques use a χ^2 likelihood function and seek to minimize the square of the residuals

$$\chi^2 = \sum_{\lambda} \left[\frac{f_{\text{D}}(\lambda) - f_{\text{M}}(\lambda|\vec{\theta}_{\star})}{\sigma(\lambda)} \right]^2 \quad (3)$$

It appears that for many techniques, $\sigma(\lambda)$ is the Poisson photon counting errors. For some data reduction pipelines, a per-pixel “sigma spectrum” that accounts for spectral extraction errors due to night sky line contamination or low signal to noise is available, and should be used instead. For more discussion of the noise sources beyond statistical Poisson errors, see §4.

ULySS ULySS (Koleva et al. 2009) is a mature package to make comparisons for individual spectra as well as unresolved stellar populations. Any spectral library can be used as a backend, including empirical spectral libraries such as the one from ELODIE (Prugniel et al. 2007). They explicitly take into account some of the necessary procedures when fitting spectra, such as continuum-multiplication by a polynomial to account for flux calibration errors and rejection of bad data or model points. They use an iterative kappa-sigma clipping method to discard “bad” points for the fit. They test that their clipping technique converges to the same answer by starting guesses located at different parameters $\vec{\theta}$ and seeing if similar lines are masked and similar best fit parameters $\vec{\theta}$ are obtained.

Mann et al. (2013) fit BT-Settl PHOENIX models to M dwarfs. Because high quality, high resolution M dwarf spectra are difficult to obtain owing to their intrinsic faintness, the development of stellar libraries for later spectral types has typically lagged behind. They use a χ^2 -like likelihood function and iterate to mask bad pixels. Unlike ULySS, however, they iterate the method against a sample of ~ 20 M dwarfs and track which pixels consistently have the high residuals (median residual amount $> 10\%$, or ten times their measurement error) across all stars. Presumably, such a consistent residual would be due to an incorrect physical constant for a certain line or molecular feature. After masking, they repeat the fitting and claim that after two iterations the masks do not change. They note that in the future, rather than selecting good and bad pixels with a binary mask, “a more nuanced weighting scheme would be to weight each interval (pixel) according to how consistent it is with the models, or test different weighting schemes to see which gives the best agreement with the bolometric temperatures. **However regions with modest agreement between the real and synthetic spectra 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.**” Thus, masking “bad regions” could very well destroy stellar information in the pursuit of a better “fit.” This is a nuanced point that masking schemes generally neglect. We aim to address later in this document.

“Bayesian” methods do exist. Shkedy et al. (2007) developed a hierarchical Bayesian approach that purportedly accounts for both statistical and systematic errors in the estimation of stellar parameters. I need to work through their math more carefully, but it seems as though they use a hierarchical model for the scatter between “true” and observed spectrum, and the unknown scatter in the relationship between $\vec{\theta}_{\star, \text{grid}}$ and the f_M . They claim to account for systematics, but it looks like these are systematics errors due to the performance of the instrument, such as a bad flux calibration. They use infrared spectra at low resolution ($R \sim 1,200$), and a weighted χ^2 function for goodness of fit. As far as I can ascertain, the actually pixel-to-pixel fitting does not account for systematic errors in regions of the model spectrum.

Schoenrich & Bergemann (2013) present a comprehensive Bayesian method that can fit low or high resolution spectra in addition to photometry and parallax data. They too, mention that they use a χ^2 likelihood. They mention using special masks for the spectra that block noisy and uninformative regions of the spectrum, however they do not go into further detail about how these masks were created or applied. Their unpublished stellar library focuses on parts of the optical spectrum which contain the most stellar information, as determined by experts. As noted by Schoenrich & Bergemann (2013), the grids “sample the wavelength windows around the spectral features important for diagnostic of FGKM stars: 3850 - 4050Å (Ca I lines), 4350 - 4450Å (G-band, CN sensitive), 4600 - 4900Å (H β), 5100 - 5300Å (Mg I triplet, main gravity diagnostics), 6400 - 6640Å (H α), 8400 - 8800Å (Ca II triplet, also used in *Gaia* and in RAVE stellar survey).”

SPC A very popular method behind the Kepler follow-up program is Stellar Parameter Classification (SPC) (Buchhave et al. 2012)). This compares high resolution optical echelle spectra ($R \sim 45,000$) to high resolution synthetic grids computed with Kurucz models by a cross-correlation method. The best fit parameters and uncertainties are calculated by the relative values of the cross-correlation coefficient. It is uncertain how statistically justified this uncertainty estimate is. While this is a pixel-to-pixel method, it uses only a small region of the optical spectrum (5000 – 5300Å) where the spectral library has been carefully tweaked¹ to match high resolution models of the Sun and Vega.

General trends Methods that are successful mask spectra lines if they are doing a full spectrum fit (although this brings up the concern of Mann et al. (2013)). Techniques which

¹changes to line centers and $\log(gf)$ values were on the table

do not mask or just use a χ^2 function are limited to a small region of the spectrum where presumably their stellar models are *good enough*. By limiting themselves to These areas can use intense scrutiny from a spectral modeller to guarantee that the models match reality.

1.3. Line fitting using spectral synthesis

This is an improvement over the line index method, since now one includes the information contained in the line profile. With quality stellar spectral models, one can identify a few well-modelled lines to fit. By choosing a few good lines to fit, one could determine stellar parameters by matching line profiles for a few select lines (Doppmann & Jaffe 2003). However, this is still subject to the need to identify promising regions of the spectrum.

MOOG² (Snedden 1973), SME³ (Valenti & Piskunov 1996), actually synthesize line profiles on the fly, compared to using a library of spectral models. This has the advantage of enabling individual tweaks to elemental abundances. And interpolation in the model atmosphere space, rather than the spectral line space. However, for many young T-Tauri stars, the spectrum is contaminated by accretion, veiling, and fractional star spot coverage, which can be a nuisance and cause systematic errors if traditional techniques are used. Matching equivalent widths. Interestingly, MOOG synthesizes a spectrum on the fly.

SME allows adjustment of oscillator strengths. We could download the synthesis code for SME and synthesize our own spectra on the fly. The biggest impediment here would be that the simplest atmospheres will not get certain lines correct. The powerful synthesis codes (PHOENIX) will get the atmospheres and lines correct, but they take a long time to run (hours) per model, and the code is not publicly available. To do a full-spectrum fit for these sorts of codes would likely be too computationally difficult. However, it would allow access to fitting in the parameter space that we want.

1.4. Stellar parameter estimation in a utopia

Although the spectral synthesis domain is the correct domain to solve for stellar parameters, the radiation transfer is too slow.

²<http://www.as.utexas.edu/~chris/moog.html>

³<http://www.stsci.edu/~valenti/sme.html>

The fitting difficulties of the previous section are a result of trying to fit data to a model in a smaller parameter space. Yes, the combination of $\vec{\theta}_{\star,\text{grid}}$ is an excellent proxy to describe a star, but in truth (and at high resolution) each star is slightly different. If we think back to first principles, the “model” that is actually producing the synthetic spectrum has many, many parameters that have been collapsed down into $\vec{\theta}_{\star,\text{grid}}$. These parameters include the atmospheric structure (relations of temperature and pressure as a function of stellar depth), individual elemental abundances (all models quote some value relative to solar, a fraction of models allow changes in alpha abundances), and atomic constants describing the opacity contribution of a line. *All* of these parameters have uncertainties in them, some of these are rather dramatic (one continually hears lamentation about the opacities and line lists. . .), yet for any given stellar model these parameters are *fixed*. There may yet be unconsidered stellar physics necessary to explain a small fraction of lines (say for example the cores of Ca lines), but the field of stellar modelling has progressed to the state that I would have the confidence to say that if a group focused all of their attention on a single line, and using only their current physical models and making tweaks to the afore-mentioned lines, they could produce a satisfactory model.

In reality, this has been done using the Kurucz models in an unpublished grid of stellar spectra, over the range 5000 – 5300 Å.

The real problem is much more rich, as is the amount of information one could extract from a high resolution optical spectrum. One would want to fit for individual abundances, marginalize over oscillator strengths.

The unfortunate reality is that sampling in the stellar model space for the full optical spectrum is not yet a computational reality. There do exist codes (MOOG) that can do this approach but only for a small (\sim a few lines) at once, and using simplifications. Through collaboration with a stellar atmospheric expert, such an approach may soon be a possibility. If the computation of a given model could be reduced to less than a minute, efficient MCMC sampling parallelized on a cluster (using `emcee`, for example). However, as it stands without a collaborator, I do not have the expertise nor the time to pursue such an approach. Therefore, we are reduced to fitting in the space of $\vec{\theta}_{\star,\text{grid}}$ and devising ways to deal with the issues.

2. General problems with the spectral library approach

Pixel-to-pixel approaches have the potential to be much more powerful than other historical stellar parameter estimation techniques. When comparing directly in the data

space, we have the ability to add in other physics as we see fit.

because we can fit multiple stars at once (say for unresolved binary or multiple systems), fit for velocity components, etc. Many of the most interesting systems for which we would want to determine stellar parameters are in multiple systems anyway, since we can dynamically determine masses for these systems and thus compare to evolutionary models. Determine veiling. And a perhaps wooden-headed belief that using the full spectrum *should* give us a more accurate and less-biased posterior estimate. Such a belief may be foolish and in reality different parts of the spectrum could give completely different parameter estimates.

Many, many methods exist for comparison of stellar spectra, including many “Bayesian” approaches in the pixel-to-pixel space. The nature of the Bayesian-ness, however, is usually limited to priors on temperature, surface gravity, and metallicity. These approaches do not explicitly state the likelihood function that they use, or if they do, it is a χ^2 function.

The more successful of these models use low or moderate resolution spectra ($R \sim 1,000$ to $10,000$ ish). Or, a grid over a narrow range of wavelength space with finely tuned atomic constants.

The true fundamental problem with this approach is that the stellar spectra are not a perfect representation of reality. Assumptions about Poisson errors break down, and fits become contaminated by systematics. There are types of errors which occur.

FIGURE: 4 Panel Figure, from left to right, showing Class 0 through III errors. Use Mg b lines, other big line near Mg b triplet, order 24 line, and then Fe I/II line complex that doesn’t fit properly.

Class I A spectral line appears in the data but not in the model spectrum over a wide range of probable stellar parameters (1000 K in T_{eff} , and 1 dex in $\log(g)$, $[\text{Fe}/\text{H}]$, and $[\alpha/\text{Fe}]$. Such a type of error is perhaps the easiest to understand and to forgive. For example, such a line could easily appear in the data due to interstellar absorption or telluric absorption, something which a stellar model could never produce. The model could also fail to reproduce a true stellar line due to a missing opacity source.

Class II A spectral line appears in the model but not in the data over a wide range of probable stellar parameters (1000 K in T_{eff} , and 1 dex in $\log(g)$, $[\text{Fe}/\text{H}]$, and $[\alpha/\text{Fe}]$. This is a more puzzling error, because presumably atomic constants would be much more likely to be *missing* than added in by accident. Presumably this is a product of bad atmospheric

structure (temperature or pressure is wrong at some point?) that excites a line when it shouldn't otherwise.

Traditional ways to model Class I and Class II errors are a form of sigma clipping. Mann et al. compares the stars to models over a wide range of parameters. One can track how these lines might be bad because Class I residuals (data - model)/sigma will always be negative (since we are dealing with absorption lines only) and not change significantly with $\vec{\theta}_{\star, \text{grid}}$, since the model can only produce continuum. Class II residuals will always be positive and should decrease with decreasing metallicity, since lower metallicity will decrease the strength of all lines.

If a Class I or II error occurs on a line, that region of the spectrum is relatively useless for determining spectral parameters. Authors like Mann et al. or Kholeva et al. find these regions and simply mask them out of the final computation. Kholeva et al., and others approach in an iterative manner. Kholeva et al. approaches are more sophisticated because they track convergence. This is a sigma clipping approach, and while we could go and model these a la Hogg 2010, the approach is not super justified, result is probably not too wrong (similar to actual sigma clipping). Given that Kholeva et al. track convergence maps, I would place a higher degree of confidence in the answer.

In our later approaches in Sec XX, how might we put a prior on $\Delta\vec{\theta}_{\star, \text{grid}}$, the range of parameters over which the residuals will be correlated? Maybe we could Gibbs sample in the same way as the bad flags. We'd have to keep a bunch of spectra with varying $\vec{\theta}_{\star, \text{grid}}$ in memory to make this comparison easy.

Class III In truth, this is just a superset of both Class I and Class II. Class III errors are when the line strengths between the data and model do not match at a $\vec{\theta}_{\star, \text{grid}}$ that fits a majority of lines. What makes Class III errors truly pernicious is that for a specific line, there does exist a set of $\vec{\theta}_{\star, \text{grid}}$ that could properly fit the line, however this parameter combination would result in bad fits for the majority of the other lines. Class I and II errors are easy to identify since the lines will always be bad across a wide range of $\vec{\theta}_{\star, \text{grid}}$.

In addition, it is difficult to determine the ensemble of lines that suffer from Class III errors, since the strength of some lines will be under-predicted while others will be over-predicted. How do we know which lines to trust? If we take an approach similar to fixing Class I and II errors, and mask out bad regions, it would be very difficult to distinguish Class III errors from *truthful* lines at an incorrect $\vec{\theta}_{\star, \text{grid}}$. For example, if one imagines using a cleaning algorithm as described for Class I and Class II errors, the identification of

Class III errors will be sensitive to the starting guesses of $\vec{\theta}_{\star, \text{grid}}$.

Because these Class III lines can still produce a significant residual spike, they tend to distort and in some cases artificially peak the posterior distribution. There is no one parameter combination that will fit all lines perfectly. Certainly there is a parameter combination that will produce a reasonable fit to 90% of the lines to within 3 sigma residuals, but there will still be a fair number of both “high” and “low” regions. These lines need to be identified and dealt with, but how?

How then, should the best fit be determined? Surely some of the remaining lines are “bad.” But in this regime, determination of which lines are bad will bias the results in a certain direction.

Also, which lines should one actually fit? The stronger lines? Did the modelers themselves give more attention to the atomic constants which produce the strong lines?

To summarize, the fundamental problem with comparing high resolution optical stellar spectra to grids of synthetic stellar spectra is that no one model $\vec{\theta}_{\star, \text{grid}}$ will fit all of the lines. Deciding how to model the lines is then the difficult part. Different groups have sigma-clipped lines that are obviously bad, and this does improve the procedure dramatically. In such a case, the procedure of sigma-clipping doesn’t gall with me too badly. However, the more difficult problem is dealing with Class III errors properly, something that no modeler has mentioned or successfully implemented in a statistically justified manner.

In a spectrum fit, many lines do reasonably fit within the Poisson errors. For consistency, we will denote these as “Class 0” lines, where there is no error.

2.1. Testing the accuracy of stellar libraries

Gaia team members are panicking about the upcoming flood of data.

Martins & Coelho (2007) compare many spectral grids to determine their weaknesses. (Sordo et al. 2011) discuss the range of stellar grids available for Gaia.

(Coelho et al. 2005) is the Coelho grid.

3. Instrumental effects and “nuisance” parameters in the limit of a perfect stellar model

The parameter vector is comprised of both stellar parameters and calibration parameters $\vec{\theta} = \{\vec{\theta}_\star, \vec{\theta}_N\}$. Stellar parameters include $\vec{\theta}_\star = \{T_{\text{eff}}, \log(g), [\text{Fe}/\text{H}], v \sin i, v_z, A_v, R_\star^2/d^2\}$ while the data vector $\vec{D} = f_D(\lambda)$ represents a high resolution spectrum. For demonstration purposes in this paper, we use a 51-order echelle spectrum from the *TRES* spectrograph at a resolution of $R = 48,000$ (6.8 km/s), which spans the full optical range (3800 – 9000 Å), though our technique is applicable to any high resolution optical/near-IR spectrum.

FIGURE showing the high resolution, raw $R = 500,000$ spectrum compared to the one convolved with a kernel down to $R = 50,000$. This shows that lines are blended, but it is important to keep in mind that we do have access to the high resolution models when talking about bad pixels or bad line-species. For example, we could inject bad lines in this space.

3.1. Processing TRES Data

We flux calibrate TRES data, but because the reduction pipeline does not produce a “sigma spectrum,” we instead use the Poisson errors on each pixel. By taking the raw, still-blazed spectrum, in units of “counts,” we can find the Signal-to-Noise ratio by,

$$S/N = \sqrt{\text{cts}} \quad (4)$$

then the Noise-to-Signal ratio is the inverse of this

$$N/S = \frac{1}{\sqrt{\text{cts}}} \quad (5)$$

then the errors on the flux-calibrated spectrum (in ergs/s/Å/cm²) is simply the flux spectrum multiplied by N/S.

3.2. Applied to a Gaussian likelihood function

Although per-pixel photon counting errors are Poisson, we can safely assume that we have enough counts that we are in the Gaussian limit, although see the discussion in §4. Then our likelihood function becomes a pixel-by-pixel χ^2 comparison between the data

spectrum f_D , and the model spectrum f_M , summed over the wavelength axis.

$$p(\vec{D}|\vec{\theta}) = \mathcal{L} \propto \exp\left(-\frac{\chi^2}{2}\right) \quad (6)$$

$$\chi^2 = \sum_{\lambda} \left[\frac{f_D(\lambda) - k(\lambda|\vec{\theta}_N)f_M(\lambda|\vec{\theta}_*)}{\sigma(\lambda)} \right]^2 \quad (7)$$

With flat priors on $\vec{\theta}_*$ and $\vec{\theta}_N$ (an assumption we will later relax) the logarithm of our posterior probability function is then

$$\ln[p(\vec{\theta}|\vec{D})] \propto -\frac{1}{2} \sum_{\lambda} \left[\frac{f_D(\lambda) - k(\lambda|\vec{\theta}_N)f_M(\lambda|\vec{\theta}_*)}{\sigma(\lambda)} \right]^2 \quad (8)$$

To generate a model spectrum given a set of parameters $f_M(\vec{\theta})$, a model spectrum is created by linearly interpolating in $T_{\text{eff}}, \log(g), [\text{Fe}/\text{H}]$ space between grid points of instrumentally broadened, oversampled PHOENIX model stellar spectra, rotationally broadened using Fourier techniques, Doppler shifted, extinction corrected, and downsampled using spline interpolation to the exact pixels of the TRES spectrum.

$\sigma(\lambda)$ is the Poisson photon counting errors. For some data reduction pipelines, a per-pixel “sigma spectrum” that accounts for spectral extraction errors due to night sky line contamination or low signal to noise is available, and should be used instead. For more discussion of the noise sources beyond statistical Poisson errors, see §4.

The prefactor $k(\lambda|\vec{\theta}_N)$ is a calibration function that aims to account for errors in the blaze-correction or flux-calibration by multiplying the model by a polynomial. The parameters describing this function are the coefficients of the polynomial. This approach of accounting for large-scale calibration errors in the spectrum is more justified than the traditional approach of normalizing to the stellar continuum. From a flux-calibration experiment where we observed spectrophotometric standards BD+28 and BD+25 multiple times, we calibrated the dispersion in (Figure?) flux-calibration function, or IRAF’s “sensitivity function.” These tests revealed that the largest source of error in flux-calibration was actually in the overall magnitude of individual order, this could be due to slit losses, poor correction due to parallactic angle, or scattered light within the instrument. These corrections are smaller than 10%. But more importantly, the corrections in the higher order terms (ie, the shape of the bandpass calibration for each order) are small ($\lesssim 3\%$). As described in §3.4, we can set priors on these coefficients. This will be the optimal mode of operation for our model. However, there exists a wealth of archival data that is

un-fluxcalibrated, since its primary purpose was for radial velocity determination of FGK stars, to which it is easy to normalize to a continuum. In this case, the scatter amongst the calibration can be as large as a factor of 2x. This prior information on the calibration parameters motivates our choice of functional parameterization for the nuisance parameters in §3.3.

3.3. Nuisance parameters

In our case, $\vec{\theta}_N = \{c_0, c_1, \dots, c_N\}$ are a set of 4 Chebyshev polynomial coefficients for each order. This amounts to a “re-fluxing” of the model following the techniques of Eisenstein et al. (2006). We apply the re-fluxing to the model rather than attempt to re-flux the data since this operation would also need to re-scale the noise, making χ^2 a complicated function of k .

In order to understand the effects of the $k(\lambda|\vec{\theta}_N)$ term, we now consider its effects on just one order of the 51-order echelle spectrum. The first four Chebyshev polynomials are

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \end{aligned}$$

In our implementation, we map the full pixel range of a specific order $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ to $x \in [-1, 1]$. By mapping pixel number in a given order, rather than wavelength, we are insensitive to wavelength shifts of the model. Then, for a given set of Chebyshev coefficients, $\vec{\theta}_N = \{c_0, c_1, c_2, \dots, c_N\}$, we have

$$k(\lambda|\vec{\theta}_N) = c_0 T_0(\lambda) [1 + c_1 T_1(\lambda) + \dots + c_N T_N(\lambda)] \quad (9)$$

$$k(\lambda|\vec{\theta}_N) = c_0 \left[1 + \sum_{i=1}^N c_i T_i(\lambda) \right] \quad (10)$$

When applied to scale the model flux f_M , this becomes

$$k(\lambda|\vec{\theta}_N) f_M = \left[1 + \sum_{i=1}^N c_i T_i(\lambda) \right] (c_0 f_M) \quad (11)$$

$$k(\lambda|\vec{\theta}_N) f_M = c_0 f_M + c_1 T_1(c_0 f_M) + \dots + c_N T_N(c_0 f_M) \quad (12)$$

This functional form of k allows c_0 to scale by large amounts while the higher order terms (denoted by c_n) are perturbations on top of the *scaled* model ($c_0 f_M$). However, see §A for an alternative model when perturbations are small. We can rewrite Equation 8 as

$$\ln[p(\vec{\theta}|\vec{D})] \propto -\frac{1}{2} \sum_{\lambda} \left[\frac{f_D(\lambda) - [1 + \sum_{i=1}^N c_i T_i(\lambda)] c_0 f_M(\lambda|\vec{\theta}_*)}{\sigma(\lambda)} \right]^2 \quad (13)$$

To simplify the following discussion, we consider a single wavelength λ_i and then generalize this to a sum over λ later. If we expand out the square in Equation 13, then for a given λ_i we have

$$\ln[p(\vec{\theta}|\lambda_i)] \propto -\frac{1}{2\sigma^2} \left[f_D^2 - 2f_D c_0 f_M \left[1 + \sum_{i=1}^N c_i T_i \right] + c_0^2 f_M^2 \left[1 + \sum_{i=1}^N c_i T_i \right]^2 \right] \quad (14)$$

where σ , f_D , f_M , and $T_{n,m}$ are all evaluated at λ_i . We can further expand this to

$$\ln[p(\vec{\theta}|\lambda_i)] \propto -\frac{1}{2\sigma^2} \left[f_D^2 - 2f_D c_0 f_M \left[1 + \sum_{i=1}^N c_i T_i \right] + c_0^2 f_M^2 \left[1 + 2 \sum_{i=1}^N c_i T_i + \sum_{i=1}^N \sum_{j=1}^N c_i T_i c_j T_j \right] \right] \quad (15)$$

To simplify this math, we can rewrite the Chebyshev coefficients and polynomials as column vectors \vec{c} and $\vec{T}(\lambda)$, respectively

$$\vec{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} \quad \vec{T}(\lambda) = \begin{bmatrix} T_1(\lambda) \\ T_2(\lambda) \\ \vdots \\ T_N(\lambda) \end{bmatrix} \quad (16)$$

then

$$k(\lambda|\vec{\theta}_N) = c_0 \left[1 + \sum_{i=0}^N c_i T_i \right] = c_0 \left[1 + \vec{T}^T \cdot \vec{c} \right] \quad (17)$$

If we let

$$\mathbf{W}(\lambda) = \vec{T} \cdot \vec{T}^T = \begin{bmatrix} T_1 T_1 & T_1 T_2 & \dots & T_1 T_N \\ T_2 T_1 & T_2 T_2 & \dots & T_2 T_N \\ \vdots & \vdots & \ddots & \vdots \\ T_N T_1 & T_N T_2 & \dots & T_N T_N \end{bmatrix} \quad (18)$$

then we have

$$\sum_{i=1}^N \sum_{j=1}^N c_i T_i c_j T_j = \vec{c}^T \cdot \mathbf{W} \cdot \vec{c} \quad (19)$$

We can rewrite Equation 15 quadratically in \vec{c} as

$$\ln[p(\vec{\theta}|\lambda_i)] \propto -\frac{1}{2\sigma^2} \left[f_D^2 - 2f_D c_0 f_M \left[1 + \sum_{i=1}^N c_i T_i \right] + c_0^2 f_M^2 \left[1 + 2 \sum_{i=1}^N c_i T_i + \sum_{i=1}^N \sum_{j=1}^N c_i T_i c_j T_j \right] \right] \quad (20)$$

$$\ln[p(\vec{\theta}|\lambda_i)] \propto -\frac{1}{2} \frac{c_0^2 f_M^2}{\sigma^2} \sum_{i=1}^N \sum_{j=1}^N c_i T_i c_j T_j + \left(\frac{-c_0^2 f_M^2 + c_0 f_D f_M}{\sigma^2} \right) \sum_{i=1}^N c_i T_i - \left(\frac{c_0^2 f_M^2 - 2c_0 f_D f_M + f_D^2}{2\sigma^2} \right) \quad (21)$$

$$\ln[p(\vec{\theta}|\lambda_i)] \propto -\frac{1}{2} \frac{c_0^2 f_M^2}{\sigma^2} \vec{c}^\top \mathbf{W} \vec{c} + \left(\frac{-c_0^2 f_M^2 + c_0 f_D f_M}{\sigma^2} \right) \vec{T}^\top \vec{c} - \left(\frac{c_0^2 f_M^2 - 2c_0 f_D f_M + f_D^2}{2\sigma^2} \right) \quad (22)$$

Because matrix multiplication is associative, we can sum $\mathbf{W}(\lambda)$ and $\vec{T}^\top(\lambda)$ across all λ , and define

$$\mathbf{A} = c_0^2 \sum_{\lambda} \frac{f_M^2(\lambda)}{\sigma^2(\lambda)} \mathbf{W}(\lambda) \quad (23)$$

$$\vec{B} = \sum_{\lambda} \frac{-f_M^2(\lambda) c_0^2 + f_D(\lambda) f_M(\lambda) c_0}{\sigma^2(\lambda)} \vec{T}(\lambda) \quad (24)$$

$$g = -\frac{1}{2} \sum_{\lambda} \frac{c_0^2 f_M^2(\lambda) - 2c_0 f_D(\lambda) f_M(\lambda) + f_D^2(\lambda)}{\sigma^2(\lambda)} \quad (25)$$

Where \mathbf{A} , \vec{B} , and g are each a function of $\vec{\theta}_\star$, $\vec{\theta}_N$, and \vec{D} . Now we can rewrite Equation 13 as

$$\ln[p(\vec{\theta}_\star, \vec{\theta}_N | \vec{D})] \propto -\frac{1}{2} \vec{c}^\top \mathbf{A} \vec{c} + \vec{B}^\top \vec{c} + g \quad (26)$$

$$p(\vec{\theta} | \vec{D}) = p(\vec{\theta}_\star, \vec{\theta}_N | \vec{D}) \propto \exp \left(-\frac{1}{2} \vec{c}^\top \mathbf{A} \vec{c} + \vec{B}^\top \vec{c} + g \right) \quad (27)$$

Here we have the full posterior probability distribution assuming flat priors. This posterior is a function of both the stellar parameters $\vec{\theta}_\star$ and the nuisance parameters $\vec{\theta}_N$. Generally, we are most interested in the stellar parameters after they have been marginalized over the nuisance parameters. This marginalization encapsulates the uncertainty due to the inference on the nuisance parameters. Because this probability function is a multi-dimensional

Gaussian in the higher order Chebyshev coefficients $\vec{c} = \{c_1, c_2, \dots, c_N\}$, we can do this marginalization analytically, and then sample from the marginalized distribution.

$$p(\vec{\theta}_\star, c_0 | \vec{D}) = \int p(\vec{\theta}_\star, c_0, \vec{c} | \vec{D}) d\vec{c} \quad (28)$$

The analytic multi-dimensional Gaussian integral with linear term (Stoof et al. 2009) yields

$$p(\vec{\theta}_\star, c_0 | \vec{D}) \propto \int \exp \left(-\frac{1}{2} \vec{c}^\top \mathbf{A} \vec{c} + \vec{B}^\top \vec{c} + g \right) d\vec{c} \quad (29)$$

$$p(\vec{\theta}_\star, c_0 | \vec{D}) \propto \sqrt{\frac{(2\pi)^N}{\det|\mathbf{A}|}} \exp \left(\frac{1}{2} \vec{B}^\top \mathbf{A}^{-1} \vec{B} + g \right) \quad (30)$$

Where N is the number of Chebyshev coefficients marginalized over, in our case $N = 3$. By doing this marginalization we have reduced the dimensionality of our posterior space from 200+ parameters (~ 7 stellar parameters plus 3 Chebyshev coefficients for each of 51 echelle orders) to ~ 60 (~ 7 stellar parameters and one c_0 for each of the 51 echelle orders). This enables a dramatic speedup in the MCMC sampling. For flux-calibrated spectra where the variation in c_0 is expected to be small (less than XX%), we can approximate k in a different manner which allows marginalization over the c_0 coefficients as well, reducing the dimensionality of the space to only the ~ 7 stellar parameters. This approach is detailed in §A.

3.4. Including Gaussian priors on nuisance parameters

Using this formalism, it is possible to incorporate prior knowledge about the degree of necessary “re-fluxing” through priors on the nuisance parameters. From our flux-calibration experiment, we determined that the overall scaling could range by 10%, while the higher order corrections were less than 3%. 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

$$p(c_0) = \frac{1}{\sqrt{2\pi}\sigma_{c_0}c_0} \exp \left(-\frac{(\ln c_0)^2}{2\sigma_{c_0}^2} \right) \quad (31)$$

$$\ln p(c_0) = \ln \left(\frac{1}{\sqrt{2\pi}\sigma_{c_0}c_0} \right) - \frac{(\ln c_0)^2}{2\sigma_{c_0}^2} \quad (32)$$

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

$$p(c_n) = \frac{1}{\sqrt{2\pi}\sigma_{c_n}} \exp\left(-\frac{c_n^2}{2\sigma_{c_n}^2}\right) \quad (33)$$

which we can express in vector form as

$$p(\vec{c}) \propto \exp\left(-\frac{1}{2}(\vec{c} - \vec{\mu})^\top \mathbf{D}(\vec{c} - \vec{\mu})\right) \quad (34)$$

where

$$\mathbf{D} = \begin{bmatrix} \sigma_1^{-2} & 0 & \dots & 0 \\ 0 & \sigma_2^{-2} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \sigma_N^{-2} \end{bmatrix} \quad (35)$$

and σ_i represents the width of the Gaussian prior on the i -th Chebyshev coefficient, in our case $\sigma_i = \sigma_{c_n} \approx 0.03$. $\vec{\mu}$ is the mean of the Gaussian prior, in nearly all cases we expect to set this to $\vec{\mu} = \{0, 0, \dots, 0\}$, unless we have prior knowledge that the flux-calibration is systematically warped in one direction. Now, we can write the posterior with priors as

$$p(\vec{\theta}_\star, \vec{\theta}_N | \vec{D}) = p(\vec{\theta}_\star, c_0, \vec{c} | \vec{D}) \propto p(\vec{D} | \vec{\theta}_\star, c_0, \vec{c}) p(\vec{c}) p(\vec{\theta}_\star) p(c_0) \quad (36)$$

$$p(\vec{\theta}_\star, \vec{\theta}_N | \vec{D}) \propto \exp\left(-\frac{1}{2}\vec{c}^\top \mathbf{A}\vec{c} + \vec{B}^\top(\vec{\theta}_\star)\vec{c} + g(\vec{\theta}_\star)\right) \exp\left(-\frac{1}{2}(\vec{c} - \vec{\mu})^\top \mathbf{D}(\vec{c} - \vec{\mu})\right) p(\vec{\theta}_\star) p(c_0) \quad (37)$$

we can expand and rearrange the argument of Equation 34 to a similar form that is quadratic in \vec{c}

$$-\frac{1}{2}(\vec{c} - \vec{\mu})^\top \mathbf{D}(\vec{c} - \vec{\mu}) = \frac{1}{2}\left(-\vec{c}^\top \mathbf{D}\vec{c} + \vec{c}^\top \mathbf{D}\vec{\mu} + \vec{\mu}^\top \mathbf{D}\vec{c} - \vec{\mu}^\top \mathbf{D}\vec{\mu}\right) \quad (38)$$

$$= -\frac{1}{2}\vec{c}^\top \mathbf{D}\vec{c} + (\mathbf{D}\vec{\mu})^\top \vec{c} - \frac{1}{2}\vec{\mu}^\top \mathbf{D}\vec{\mu} \quad (39)$$

then we can rewrite

$$\mathbf{A}' = \mathbf{A} + \mathbf{D} \quad (40)$$

$$\vec{B}' = \vec{B} + (\mathbf{D}\vec{\mu})^\top \quad (41)$$

$$g' = g - \frac{1}{2}\vec{\mu}^\top \mathbf{D}\vec{\mu} \quad (42)$$

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

$$p(\vec{\theta}_*, \vec{\theta}_N | \vec{D}) = p(\vec{\theta}_*, c_0, \vec{c}) \propto \exp\left(-\frac{1}{2}\vec{c}^\top \mathbf{A}' \vec{c} + \vec{B}'^\top \vec{c} + g'\right) p(\vec{\theta}_*) p(c_0) p(\vec{c}) \quad (43)$$

This probability function appears in the code as `lnprob_lognormal` and has ~ 7 stellar dimensions and ~ 200 nuisance dimensions.

Using the previous multi-dimensional integral, we can marginalize over the \vec{c} including the Gaussian priors

$$p(\vec{\theta}_*, c_0 | \vec{D}) \propto \sqrt{\frac{(2\pi)^N}{\det|\mathbf{A}'|}} \exp\left(\frac{1}{2}\vec{B}'^\top \mathbf{A}'^{-1} \vec{B}' + g'\right) p(\vec{\theta}_*) p(c_0) \quad (44)$$

where N is the number of Chebyshev coefficients marginalized over, in our case $N = 3$. This probability function appears in the code as `lnprob_lognormal_marg` and has ~ 7 stellar dimensions and ~ 50 nuisance dimensions. If we increase the permissible degree of re-fluxing by increasing σ_i , this will have the effect of smearing out or widening the posterior on a given $\vec{\theta}_*$.

3.5. Sampling a un-flux calibrated model

This invalidates the previous section on normalization. However, for corrections of less than XX%, this approximation is better than XX% accurate.

For the flux-calibration exercise, the overall spread in c_0 across all orders should not be more than $\sigma_{c_0} \leq 2$, or the one-sigma deviation of c_0 is between $0.125 \leq c_0 \leq 8.0$.

The plot in plots/priortests shows that the log-normal prior is the correct choice. Since we care most about the inference on F, we want to know that the choice on prior of c_0 is not biasing the final outcome, while a Gaussian prior might if σ_{c_0} was large. How to quantify this error and create a dividing line? For 5% error? In peak? Do these plots need to be in the paper?

Compare distributions by doing a K-S test on the cumulative distributions between Gaussian and Log-normal?

3.6. Hierarchical modelling

Can have a c_1 , etc, that are marginalized over but we sample c_0 in a hierarchical manner which informs the overall flux-level.

3.7. Sampling with emcee

Citation to DFM (Foreman-Mackey et al. 2012), GW (Goodman & Weare 2010), and CosmoHAMMER (Akeret et al. 2013).

4. Moving beyond a perfect stellar model

4.1. Additional sources of noise beyond Poisson

Presumably we shouldn’t have to add an additional source of “noise” to the fit, because our nuisance parameters will find an exact flux calibration.

Interpolation errors. Quantified via Husser et al. Ph.D. thesis as well as custom tests.

4.2. Noise, Masking, Residuals, and Alternative likelihood functions

In the future, we could modify our likelihood function to use a Student-t statistic instead of a Gaussian, such that it would be more robust in the presence of outlier pixels, however this does not seem necessary at the moment.

4.3. Masking

We must necessarily mask certain wavelength regions of the T Tauri spectra that are contaminated (one might also say “made more interesting”) by the astrophysical realities of accretion or stellar spots (such as the Balmer lines, which are in emission). We also accept the assumption that rotation or accretion onto the star does not fundamentally alter the structure of the model atmosphere that generates the model spectrum and that a spectral comparison of the “clean” regions of the spectrum is valid.

Section on how to choose masked regions, testing the PHOENIX grid to see where the

derivative is highest and thus gives the best leverage on stellar parameters.

Iterative masking to deal with Class I and II errors a la ULYSS. They use a $\kappa - \sigma$ routine, where they iteratively sigma clip (Website link).

5. Gallery of likelihood functions

If you do a straight up χ^2 fit to the data, you can get posteriors that look like ± 1 K and ± 0.01 dex in gravity and metallicity. This would be better than that obtained by astroseismology! Clearly there is something going wrong here.

Lorentzian, linear combination of Gaussians, Sivia w/ prior on sigma. χ_Q^2 model.

These likelihood functions seem to be an improvement over the straight χ^2 function, in the sense that they give parameter estimates that are

5.1. Bad data model

The bad data model (Press 1997; Hogg et al. 2010) is a powerful framework for forward-modelling of data, allowing the modeler to account for the probability of outliers.

Now, we have a q_i for each data point, such that 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 data points are flagged as good and the bad data points are flagged as bad, the likelihood will be maximized. Written out explicitly, this is

$$\mathcal{L} = p(\mathbf{D}|m, b, \{q_i\}_{i=1}^N, Y_b, V_b) = \prod_{i=1}^N \left[\frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(y_i - mx_i - b)^2}{2\sigma^2} \right] \right]^{q_i} \times \left[\frac{1}{\sqrt{2\pi(V_b + \sigma_{y_i}^2)}} \exp \left[-\frac{(y_i - Y_b)^2}{2(V_b + \sigma_{y_i}^2)} \right] \right]^{1-q_i} \quad (45)$$

using a Bernoulli prior

$$p(q_i|P_b) = [1 - P_b]^{q_i} P_b^{[1-q_i]} \quad (46)$$

$$p(\{q_i\}_{i=1}^N | P_b) = \prod_{i=1}^N [1 - P_b]^{q_i} P_b^{[1-q_i]} \quad (47)$$

For this data point, we can integrate (in this case, discrete sum) over the two states of q_i : $q_i = 1$ and $q_i = 0$.

$$\begin{aligned} p(m, b, P_b, Y_b, V_b | D_i) &= \int p(m, b, q_i, P_b, Y_b, V_b | D_i) dq_i \\ &= p(m, b, q_i = 1, P_b, Y_b, V_b | D_i) + p(m, b, q_i = 0, P_b, Y_b, V_b | D_i) \\ &= \left[p_{\text{good}}(D_i | m, b) p(q_i = 1 | P_b) + p_{\text{bad}}(D_i | Y_b, V_b) p(q_i = 0 | P_b) \right] \times p(m, b, P_b, Y_b, V_b) \end{aligned} \quad (48)$$

$$\begin{aligned} p(m, b, P_b, Y_b, V_b | \mathbf{D}) &\propto \prod_{i=1}^N \left[[1 - P_b] p_{\text{good}}(D_i | m, b) + P_b p_{\text{bad}}(D_i | Y_b, V_b) \right] \times p(m, b, P_b, Y_b, V_b) \\ &= \prod_{i=1}^N \left\{ \frac{1 - P_b}{\sqrt{2\pi\sigma_{yi}^2}} \exp \left[-\frac{(y_i - mx_i - b)^2}{2\sigma_{yi}^2} \right] + \frac{P_b}{\sqrt{2\pi(V_b + \sigma_{yi}^2)}} \exp \left[-\frac{(y_i - Y_b)^2}{2(V_b + \sigma_{yi}^2)} \right] \right\} \times p(m, b, P_b, Y_b, V_b) \end{aligned} \quad (49)$$

In it's simplest form (Hogg 2010), the bad data model results in a linear combination of Gaussians.

A benefit of the bad data model over a chosen likelihood function is that it is motivated. It may still result in the linear combination of Gaussians (nice), but preserves real parameter correlations. You are not forcing the same mean.

5.2. Spatial correlations of pixels

We can do more. The bad model of the previous section assumes that each data point is independent. For example, the spectral lines are actually bad, and this means that if a single line is wrong, not just the core of the line (which may be a 10 - 20 sigma residual) will be wrong, but also the wings of the line, which may be only 5 or 3 sigma wrong.

Now, let's come up with a model which includes spatial (wavelength) correlations in the bad data model.

How could we accomplish this? If we identify one point as bad, then the flags around it will have a correlation length, ie, the prior on q_i will be bad. This means that for however many nearby pixels that will be bad (most likely a constant amount for uncrowded lines), then this means that we could still integrate over the q_i s, it’s just that we would now have to take a contribution from the nearest 5 pixels, say. That central pixel would then mark the “line” that is bad.

That is one way to mark the “badness of the line,” that the q_i are correlated. But we also know that the y_i will be correlated too.

This correlation length will probably depend on the type of star, and the contribution of the band. For example, if they’re simple FGK stars, then the bad lines might be singular. But if they are M stars, then there is a chance that there might be a large “pseudo-continuum” region that might be incorrect.

Once we have identified the center of the bad line, then there will be a decrease in the nearby pixels, similar to a Gaussian shape. Presumably we could come up with a correlation in this space too. (or, it could simply be that the correlation between y_i is flat in the case of an M star.

This might actually save the marginalization. We could still do the backwards “what is the probability a given line is bad.”

For Class I and II errors, we could have a hyper parameter describing the width of each line, and another hyper parameter describing the (distribution of) depth. These could also be sampled in.

For Class III errors, describing how line strengths are off makes sense for stars with continuum, but as we move to later type stars (K stars) that do not have a nice continuum, we may run into trouble. For the later stars (M dwarfs), there might be a whole region that is incorrect.

There are different ways to describe a correlation over pixels. Gaussian random fields. Autocorrelation function.

5.3. Simple correlation between pixels

For a simple correlation, say that if a given q_i is bad, then there is also a high probability that the neighboring pixels are also bad. Say we know from inspection that it seems as though bad lines knock out 5 pixel chunks.

Caveat: I think this assumes that bad lines will not occur within the same spread of each other. Maybe we should include a P_B prior on the good counts too?

$$p(q_i = 0 | q_{i\pm 5} = 0) = P_B \quad (50)$$

$$p(q_i = 0 | q_{i\pm 5} = 0) = 1/5 \quad (51)$$

$$q_{i\pm 4} = 2/5 \quad (52)$$

$$q_{i\pm 3} = 3/5 \quad (53)$$

$$q_{i\pm 2} = 4/5 \quad (54)$$

$$(55)$$

This might need more work with P_B .

Then the probability functions would be

$$p(D_{i+j} | \vec{\theta}, q_i = 0) = \prod_{j=-4, \neq 0}^4 [p(D_{i+j} | \vec{\theta}, q_{i+j} = 0) p(q_{i+j} = 0 | q_i = 0) + p(D_{i+j} | \vec{\theta}, q_{i+j} = 1) p(q_{i+j} = 1 | q_i = 0)] \quad (56)$$

$$p(D_{i+j} | \vec{\theta}, q_i = 1) = \prod_{j=-4, \neq 0}^4 [p(D_{i+j} | \vec{\theta}, q_{i+j} = 0) p(q_{i+j} = 0 | q_i = 1) + p(D_{i+j} | \vec{\theta}, q_{i+j} = 1) p(q_{i+j} = 1 | q_i = 1)] \quad (57)$$

The marginalization could still be done analytically, but now the likelihood function involves a correlation between adjacent pixels. Could we use a sparse array for this? Or, we could evaluate all of the pixels, and then have the final likelihood function involve some correlation terms between the lnprobs evaluated at each pixel. We could do this using the logarithm so that everything is additive.

Or, what if we evaluated the FFT of the $\log(p(D_i))$ and just the autocorrelation in Fourier space?

5.4. Correlation between y_i

Not only will the flags be spatially correlated (if a point is bad, the nearby ~ 5 points are also likely to be bad) but for OBAFGK stars, the lines are likely to be Gaussian shape. That is if q_i specifies the center of the line, then the neighboring y_{i+1} or y_{i-1} will be of lower residual height. This means that the inside of the likelihood function should also have spatial correlations.

Could this correlation be described by a Gaussian random process?

5.5. Different flags and weights

Instead of a good or bad flag, what if we had a flag that would identify which class of error the line became? For example we could have the option of $q_i = \{0, 1, 2, 3\}$, where $q_i = 0$.

How do we account for the fact that some parts of the spectrum carry much more information about $\vec{\theta}_{\star, \text{grid}}$ than other parts of the spectrum? Is it important to weight this parts beforehand? Or should this contribution to the posterior naturally weight itself, the way certain points might heavily contribute to the slope and intercept of a linear fit?

If so, we could imagine having a Gibbs sampler on the weights of each pixel.

Proper Class III modeling is very important, following the discussion of (Mann et al. 2013) Mann et al, and see recent example from Rajpurohit 2103. Some features will not fit perfectly, especially at the lower temperature scale. However, it's important to still take these features into account with the modelling. They also use a χ^2 likelihood function, but perhaps even less appropriate is that they draw error bars by saying that the χ^2 value changes by 5%. In reality, they are checking the fits by eye and saying whether or not a spectrum matches. This approach is no more justified than a by-eye spectral classification.

5.6. Sampling such a model

If our bad model is sufficiently complex that we are not able to analytically marginalize over the q_i , it is still possible to sample in this huge parameter space. A stellar model with flags for each pixel would have $\vec{\theta}_{\star} + N$ parameters, where for a typical optical spectrum $N > 100,000$. Even using an ensemble sampler, sampling in this space would take a tremendous amount of time to converge.

An alternative way to approach this problem is to do Gibbs sampling in the flags. It may be possible to do Gibbs sampling where we draw from the conditional probability distribution, or at the very worst we could still do a Gibbs sampler with a Metropolis-Hastings backend, where we propose a jump in a single q_i at a time. Such a sample requires none of the previous heavy lifting (spectral grid interpolation, FFT, or pixel interpolation), and so even though there are $\sim 100,000$ flags to sample, this is likely to be computationally feasible.

6. Injection of fake lines

An alternative approach is instead of modelling the systematically wrong regions of the spectrum, actually add in lines to the spectrum. This can be thought of as correcting the models. We could add in a set number of lines.

Presumably, each line would be a Gaussian parameterized by λ_0 , strength, and FWHM. Strength could be described by a hyper prior, and FWHM described by a hyper parameter, since all lines should have pretty much the same FWHM (in velocity space).

In later type spectra, this might actually need to be a larger continuum feature that is added.

We could penalize the model fits by having a hyperprior on the number of bad lines that can be created.

In theory, it is possible to fit the profiles of only certain elements to determine the abundances of certain elements (see citation in conference proceedings of Husser).

It may be possible to use a linelist from the group that produces the PHOENIX spectra, which describes the element that gave the highest contribution to the opacity in that bin. Now, if the vanadium lines are bad, then we could have a prior that the other vanadium lines might also be bad. We could assemble a piecewise spectrum, where certain pixels are drawn from a different combination of $\vec{\theta}_{\star, \text{grid}}$, and these pixels had an associated normalization constant so that the piecewise transition was smooth (if, say the bolometric temperature was different).

Class III errors might be difficult to fit in this approach. However, if the line width does not change dramatically, then we should be able to account for things this way.

7. Validation tests

Comparison to main sequence stars of HAT-P-9, and WASP-14, from the TRES archive (Torres et al. 2012).

Comparisons using HIRES data?

IC would like to graciously acknowledge Gregory Green, Doug Finkbeiner, and Daniel Eisenstein for many fruitful discussions about statistics and spectroscopy. This research made use of Astropy, a community-developed core Python package for Astronomy (Astropy Collaboration et al. 2013).

A. 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 ~ 7 stellar parameters, $\vec{\theta}_\star$.

$$k(\lambda|\vec{\theta}_N) = c_0 T_0(\lambda) + c_1 T_1(\lambda) + \dots + c_N T_N(\lambda) \quad (\text{A1})$$

$$k(\lambda|\vec{\theta}_N) = \sum_{i=0}^N c_i T_i(\lambda) \quad (\text{A2})$$

Following §3.3

$$\ln[p(\vec{\theta}|\vec{D})] \propto -\frac{1}{2} \sum_{\lambda} \left[\frac{f_D(\lambda) - [\sum_{i=0}^N c_i T_i(\lambda)] f_M(\lambda|\vec{\theta}_\star)}{\sigma(\lambda)} \right]^2 \quad (\text{A3})$$

$$\ln[p(\vec{\theta}|\lambda_i)] \propto -\frac{1}{2\sigma^2} [f_D^2 - 2f_D f_M \sum_n c_n T_n + f_M^2 \sum_n \sum_m c_n T_n c_m T_m] \quad (\text{A4})$$

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

$$\vec{c} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} \quad \vec{T}(\lambda) = \begin{bmatrix} T_0(\lambda) \\ T_1(\lambda) \\ \vdots \\ T_N(\lambda) \end{bmatrix} \quad (\text{A5})$$

$$k(\lambda|\vec{\theta}_N) = \sum_{i=0}^N c_i T_i = \vec{T}^\top \cdot \vec{c} \quad (\text{A6})$$

$$\mathbf{W}(\lambda) = \vec{T} \cdot \vec{T}^\top = \begin{bmatrix} T_0 T_0 & T_0 T_1 & \dots & T_0 T_N \\ T_1 T_0 & T_1 T_1 & \dots & T_1 T_N \\ \vdots & \vdots & \ddots & \vdots \\ T_N T_0 & T_N T_1 & \dots & T_N T_N \end{bmatrix} \quad (\text{A7})$$

$$\sum_n \sum_m c_n T_n c_m T_m = \vec{c}^\top \cdot \mathbf{W} \cdot \vec{c} \quad (\text{A8})$$

$$\ln[p(\vec{\theta}|\lambda_i)] \propto -\frac{1}{2\sigma^2} [f_D^2 - 2f_D f_M \vec{T}^\top \vec{c} + f_M^2 \vec{c}^\top \mathbf{W} \vec{c}] \quad (\text{A9})$$

$$\mathbf{A} = \sum_{\lambda} \frac{f_M^2(\lambda)}{\sigma^2(\lambda)} \mathbf{W}(\lambda) \quad (\text{A10})$$

$$\vec{B}^\top = \sum_{\lambda} \frac{f_D(\lambda) f_M(\lambda)}{\sigma^2(\lambda)} \vec{T}^\top(\lambda) \quad (\text{A11})$$

$$g = -\frac{1}{2} \sum_{\lambda} \frac{f_D(\lambda)}{\sigma^2(\lambda)} \quad (\text{A12})$$

$$(\text{A13})$$

Where \mathbf{A} , \vec{B} , and g are each a function of $\vec{\theta}_*$, $\vec{\theta}_N$, and \vec{D} . If we include Gaussian priors, then we have

$$\mathbf{A}' = \mathbf{A} + \mathbf{D} \quad (\text{A14})$$

$$\vec{B}' = \vec{B} + (\mathbf{D} \vec{\mu})^\top \quad (\text{A15})$$

$$g' = g - \frac{1}{2} \vec{\mu}^\top \mathbf{D} \vec{\mu} \quad (\text{A16})$$

where now \mathbf{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 $\vec{\mu} = \{1, 0, \dots, 0\}$. The full posterior probability function becomes

$$p(\vec{\theta}_*, \vec{\theta}_N | \vec{D}) = p(\vec{\theta}_*, \vec{c}) \propto \exp \left(-\frac{1}{2} \vec{c}^\top \mathbf{A}' \vec{c} + \vec{B}'^\top \vec{c} + g' \right) p(\vec{\theta}_*) p(\vec{c}) \quad (\text{A17})$$

Using the previous multi-dimensional integral, we can marginalize over all \vec{c} including the Gaussian priors

$$p(\vec{\theta}_*|\vec{D}) \propto \sqrt{\frac{(2\pi)^N}{\det|\mathbf{A}'|}} \exp\left(\frac{1}{2}\vec{B}'^T \mathbf{A}'^{-1} \vec{B}' + g'\right) p(\vec{\theta}_*) \quad (\text{A18})$$

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

$$k(\lambda|\vec{\theta}_N) = c_0 T_0 + c_1 T_1 + \dots + c_N T_N \quad (\text{A19})$$

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 $0.7 \leq c_0 \leq 1.42$. Beyond this, one should switch to this framework, for example to correct for un-flux calibrated data.

REFERENCES

- Akeret, J., Seehars, S., Amara, A., Refregier, A., & Csillaghy, A. 2013, CosmoHammer: Cosmological parameter estimation with the MCMC Hammer, 1212.1721, *astro-physics Source Code Library*
- Astropy Collaboration et al. 2013, *A&A*, 558, A33
- 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
- Coelho, P., Barbuy, B., Meléndez, J., Schiavon, R. P., & Castilho, B. V. 2005, *A&A*, 443, 735, *astro-ph/0505511*
- D’Antona, F., & Mazzitelli, I. 1997, *Mem. Soc. Astron. Italiana*, 68, 807
- Doppmann, G. W., & Jaffe, D. T. 2003, *AJ*, 126, 3030, *astro-ph/0308521*

- 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
- 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
- Koleva, M., Prugniel, P., Bouchard, A., & Wu, Y. 2009, *A&A*, 501, 1269, 0903.2979
- Lee, Y. S. et al. 2008, *AJ*, 136, 2022, 0710.5645
- Mann, A. W., Gaidos, E., & Ansdell, M. 2013, *ApJ*, 779, 188, 1311.0003
- Martins, L. P., & Coelho, P. 2007, *MNRAS*, 381, 1329, 0705.1994
- Newton, E. R., Charbonneau, D., Irwin, J., Berta-Thompson, Z. K., Rojas-Ayala, B., Covey, K., & Lloyd, J. P. 2014, *AJ*, 147, 20, 1310.1087
- Paletou, F., Trouilhet, J.-F., & Boehm, T. 2014, ArXiv e-prints, 1401.1083
- Press, W. H. 1997, in *Unsolved Problems in Astrophysics*, ed. J. N. Bahcall & J. P. Ostriker, 49–60, arXiv:astro-ph/9604126
- Prugniel, P., Soubiran, C., Koleva, M., & Le Borgne, D. 2007, *VizieR Online Data Catalog*, 3251, 0
- Recio-Blanco, A., Bijaoui, A., & de Laverny, P. 2006, *MNRAS*, 370, 141, astro-ph/0604385
- Santos, N. C. et al. 2013, *A&A*, 556, A150, 1307.0354
- Schoenrich, R., & Bergemann, M. 2013, ArXiv e-prints, 1311.5558
- Shkedy, Z., Decin, L., Molenberghs, G., & Aerts, C. 2007, *MNRAS*, 377, 120, arXiv:astro-ph/0701449
- Snedden, C. A. 1973, PhD thesis, THE UNIVERSITY OF TEXAS AT AUSTIN.
- Sordo, R. et al. 2011, *Journal of Physics Conference Series*, 328, 012006
- 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
- Valenti, J. A., & Piskunov, N. 1996, *A&AS*, 118, 595