A Method for the spectroscopic inference of fundamental stellar parameters

May 6, 2014

Ian Czekala[1] et al.

iczekala@cfa.harvard.edu

1. Introduction

- Where our technique fits in the ecosystem of stellar methods.
- Applicable to any kind of spectrum (long-slit, echelle, infrared, flux-calibrated or not)
- Examples of fields where accurate and unbiased stellar parameters are crucial. Exoplanets. T Tauri stars.

2. Method

SA: This section needs a true introduction of its own. So far, you just sort of fly right into the details. The introduction should basically summarize the underlying approach you are going to take, and then literally lay out a basic roadmap for the following subsections. Some suggested paragraphs below.

SA: (1) describe the *philosophy* behind the approach: you want to make inferences on fundamental stellar parameters using the vast amount of information available in a spectrum, but at the same time you want to acknowledge and pro-actively account for both intrinsic parametric degeneracies and any practical covariances introduced into the data and the way its treated. Ideally, this can be done in a flexible, modular way to easily incorporate additional complexity to treat different kinds of problems. By casting the problem in a specific (Bayesian) framework, you can do that while also simultaneously learning in a data-driven way about how the underlying models of stellar spectra might be improved. [Note that this paragraph might seem repetitive in the paper, but it sort of encapsulates all the key points so it merits reinforcement.]

SA: (2) now give an overview of the basic “steps" involved in the process; in essence this is a $\sim$1-2 sentence summary of each subsequent subsection. This is to give the reader a “big picture" overview of the methodology. If we exclude this, then the following subsections become very confusing and incoherent, even for an expert in the subject. At the same time, you want to be specific enough that its clear what you are going to do: (a) for a given set of stellar parameters, you generate (somehow) a synthetic model spectrum (Section 2.1); (b) you then post-process that spectrum for realism (using additional parameters) and to match the data (using nuisance parameters) (Section 2.2); (c) you then perform a pixel-by-pixel

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

comparison with the data using a prescribed likelihood function (Section 2.3) and a parametric treatment of the covariance matrix (Section 2.4); and then (d) you iterate this procedure in an MCMC simulation to explore the posterior PDF (Section 2.5). At the end, you want to be specific about what you "get" from this procedure: robust estimates of the stellar parameters (and associated nuisance parameters), and a characterization of the covariance matrix that could (in principle) be used to identify and quantify the key discrepancies between the data and the models.

2.1. Synthetic Model Spectra (SA: note change)

SA: This subsection needs to make a little more effort to be generic first. Your initial instinct is fine: first introduce the (three) fundamental parameters you care about (but note that these could be expanded in principle, to include things like B-fields, etc.). But then, we do not want to state a "preferred" way of generating a synthetic model spectrum. Instead, it would be interesting to make a few comments here harking back to the different approaches you have outlined (or will outline) in Sect 1 – you can (a) run a stellar structure + radiative transfer code to generate a spectrum corresponding to these parameters (brute force; impractical computationally); (b) interpolate from a grid of stellar structures and run a simple radiative transfer code (i.e., SME); or (c) interpolate from a library of synthetic spectra (i.e., SPC, maybe others...). Comment on the trade-off for speed versus accuracy (very briefly), and then say that we are going to focus here on the latter approach (although fundamentally the methodology is applicable to all three equally).

Typically, the stellar properties we most desire are effective temperature $T_{\rm eff}$, radius or surface gravity $\log g$, and metallicity [Fe/H]. Several high-quality libraries of synthetic spectra now exist (PHOENIX, Kurucz, more being developed for GAIA) and are parameterized by these fundamental stellar parameters, which together, we call $\vec{\theta}_{\star,\rm grid}$. Synthetic libraries provide high-resolution spectra which span a range of $\vec{\theta}_{\star,\rm grid}$ covering the main sequence of spectral types. The library is typically specified on a grid of equal spacing in $T_{\rm eff}$, $\log g$, and [Fe/H]. SA: Note the re-arrangement here (below); I think this subsection should deal only with "obtaining" a full-resolution synthetic model spectrum with the key stellar parameters. So, I've moved stuff about $v \sin i$, etc. into Sect 2.2.

SA: These sentences should be recast and moved into the Sect 2 introduction. Using a synthetic stellar library as a backend and applying post-processing techniques, we attempt to forward-model the observed stellar spectra in order to determine the best-fit $\vec{\theta}_{\star}$. Because properly sampled spectra have correlated pixels, we present a framework which self-consistently models these correlations.

Creating a new spectrum with a specific $\vec{\theta}_{\star,\rm grid}$ requires either synthesizing a new spectrum using radiative transfer codes or interpolating from nearby grid points in the library. Because it is computationally expensive to synthesize a spectrum at high resolution over a wide wavelength range, we choose to interpolate. Spline interpolation for spectra (Husser 2012).
TODO: Implement band-limited interpolation from the synthetic grid using spline interpolation. Caution: As is, 100K/0.5 logg spacing in $\vec{\theta}_{\star,\rm grid}$ may not be Nyquist sampling. SA: I think this is an important thing to do *now*.

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
$\vec{\theta}_\star$	$\{\vec{\theta}_{\star,\mathrm{grid}}, \vec{\theta}_{\star,\mathrm{post}}\}$
$f_{\lambda,\mathrm{inst}}(\lambda)$	data spectrum
$f_{\lambda,\mathrm{synth}}(\lambda)$	synthetic spectrum
$c_0, c_1, \ldots$	Chebyshev polynomial coefficients for residual flux calibration
$\vec{\theta}_{\mathrm{N}}$	the set of nuisance parameters composed of $\{c_0, c_1, \ldots, c_N\}$
$\vec{\theta}$	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, $D(\lambda_i)$
D	data vector comprised of all D_i, $i = \{0, \ldots, N\}$
M_i	model flux for a given pixel, $M(\lambda_i \vert \vec{\theta})$
M	model vector comprised of all M_i, $i = \{0, \ldots, N\}$
σ_i	Poisson noise for a given pixel i
R_i	residuals $D_i - M_i$
R	residual vector $D - M$
C	covariance matrix
k_{global}	global covariance kernel
k_{region}	regional covariance kernel

Table 1: Nomenclature used in this document SA: Obviously this table does not belong here...

TODO: Investigate a "Bayesian Emulator" for using Gaussian processes for interpolating from a simulation grid. SA: I do not think this is essential, so it should be put off until the paper is basically done, if even then.

2.2. Post-Processing

SA: You need to incorporate these statements (moved from previous subsection) somewhere into the paragraph below. In addition to these fundamental parameters, a star also has several observed properties that are a function of its kinematics, geometry, and location in our galaxy: projected rotational velocity $v \sin i$, line of sight velocity v_z, total extinction A_V, and solid angle Ω. We call these secondary parameters $\vec{\theta}_{\star,\mathrm{post}}$, because we can model their effects on the synthetic spectra in a "post-processing" step by convolution with an appropriate kernel or multiplication with a smooth function. Together, we call $\vec{\theta}_\star = \{\vec{\theta}_{\star,\mathrm{grid}}, \vec{\theta}_{\star,\mathrm{post}}\}$.

Libraries of synthetic spectra are generally computed at high resolution ($R \geq$100,000), sampled with

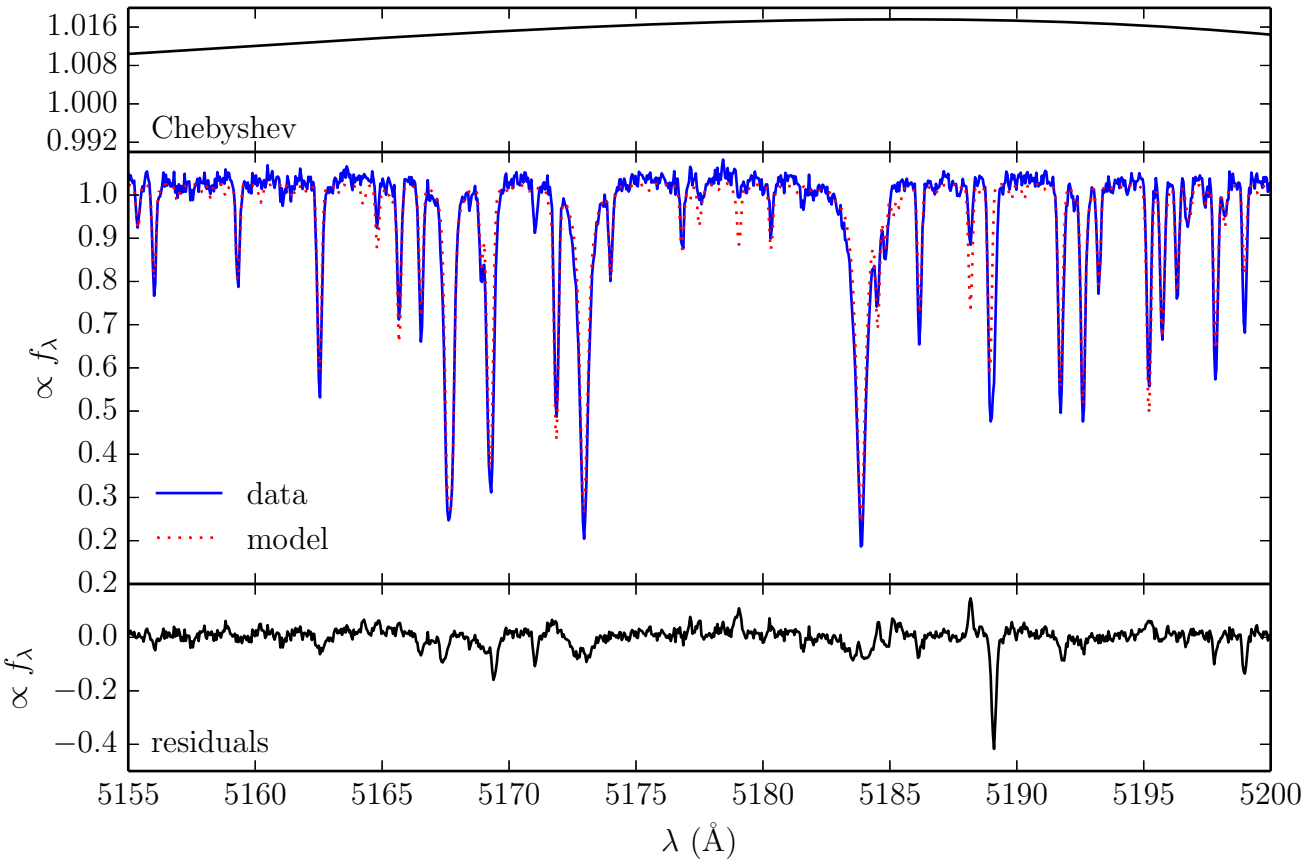

Fig. 1.— **Top**: Chebyshev polynomial which modifies the model spectrum to account for inaccuracies in the flux-calibration. **Middle**: The data spectrum and model spectrum, after it has been interpolated, post-processed, and multiplied by the Chebyshev polynomial. **Bottom**: Residuals from the model fit. Note the large residual at λ5189Å due to a missing opacity source from Ca I.
TODO: Model spectrum will be updated once I properly burn in the chain. Do these line styles look OK? I was going for something legible that might also print well in B/W.

many pixels per resolution element, and do not include any rotational broadening or account for any instrumental effects. In order to transform a raw synthetic spectrum interpolated from the grid into one that matches the spectrum of a real star, we must post-process the spectrum to account for these secondary effects. The projected rotational velocity of the star, parameterized by $v \sin i$, broadens spectral line profiles and is mathematically described by convolution with a kernel $g_{v \sin i}(\lambda)$. UV, optical, and infrared spectra are acquired using a spectrograph which disperses light onto a CCD, defining a specific resolution and sampling rate for the spectrum. Because the resolution of the spectrograph and spacing of the pixels are different from

the synthetic spectral library, we must convolve the raw spectrum with the line spread function (LSF) of the instrument $g_{\rm LSF}(\lambda)$ and resample to the exact pixels of the CCD. In wavelength space, the $v \sin i$ and LSF operations are represented by the convolution of the synthetic spectrum with these two kernels

$$f_{\lambda,\rm inst}(\lambda) = g_{v \sin i}(\lambda) \otimes g_{\rm LSF}(\lambda) \otimes f_{\lambda,\rm synth}(\lambda) \tag{1}$$

Using the convolution theorem, we can rewrite these operations as multiplications in Fourier space

$$F_{\lambda,\rm inst}(s) = G_{\rm LSF}(s) G_{v \sin i}(s) F_{\lambda,\rm synth}(s) \tag{2}$$

where $G \leftrightarrow g$ and $F \leftrightarrow f$ are Fourier transform pairs. In order to use the fast Fourier transform (FFT) to execute these operations using discrete samples of the synthetic spectrum $f_{\lambda,\rm synth}$, we must first resample $f_{\lambda,\rm synth}$ so that it is on a uniform grid. In the case of a spectrum, the natural uniform grid is one that is equally sampled in velocity space, such there is an equal velocity shift Δv between pixels. This results in a wavelength grid that is linearly sampled in the logarithm of wavelength. Therefore, the Fourier coordinate s is the number of cycles per sampling interval, having units of inverse velocity [s/km]. Next, we multiply the Fourier transform of the synthetic spectrum by the Fourier transforms of the rotational velocity and line spread function kernels. Lastly, we do an inverse FFT to transform the modified spectrum $F_{\lambda,\rm inst}$ back to wavelength space, where it is sampled at the wavelength locations of the pixels in the detector ($f_{\lambda,\rm inst}$). When resampling the spectrum, we are careful to ensure band-limited interpolation using splines in order to prevent introducing non-physical high frequency structure into the spectrum.

SA: It seems odd to me to do the extinction correction here, after all the work on velocities. I suppose it does not really matter, but it would seem more harmonious if you finished discussion of all velocity-related manipulation and then applied both the extinction correction and the overall luminosity (or solid angle) scaling after that. So, maybe re-arrange a bit for a more logical flow (regardless of what the code itself is doing). Finally we apply corrections for interstellar extinction and Doppler shift. Interstellar extinction attenuates the spectrum by an amount A_λ, which is wavelength-dependent. We use the (name here) dust-extinction model, which is parameterized by extinction in the V photometric band, A_V. We correct for a radial velocity difference v_z between the star and the earth by Doppler shifting the model spectrum.
TODO: Doppler shifting the wavelengths could be implemented as a phase-shift in Fourier space, but it's pretty fast as-is. SA: Is there a reason that would be better? If this is not a bottleneck, I would ignore this.

TODO: We might try implementing the convolution in wavelength space, rather than the FFT. This would allow the LSF to change with wavelength.
Pixel width effects are not important as long as the LSF of the instrument is adequately sampled ($\gtrsim 3$ pixels across the LSF). If our spectrum were not properly sampled, we would need to additionally convolve with a boxcar the width of the pixels. SA: I interpret this as a comment to yourself...either integrate into the text in a clear way, or remove.

SA: I feel like this (important) part of the section is rushed into...it is lacking some kind of introductory, pedagogical lead-in to make the reader recognize its power and flexibility. You are essentially touching on the big issue of parameterizing your calibration. Now, we don't want to go down that path in detail, but

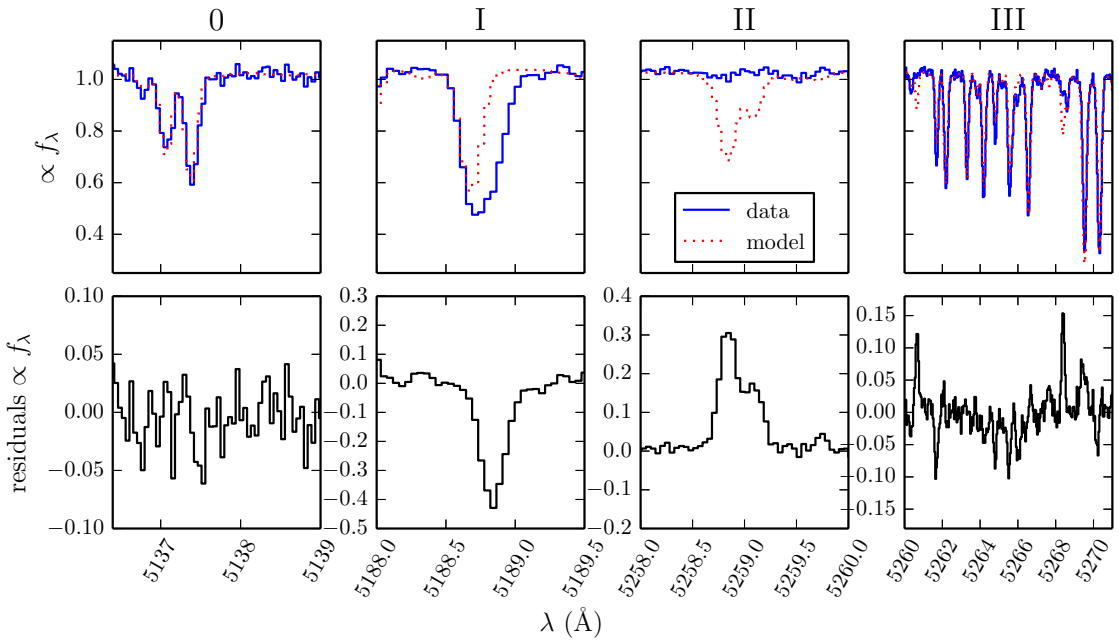

Fig. 2.— The variety of residual behaviour, depending on the quality of model fit. From left to right: **Class 0** covariance results from slight model mismatch, and correlates nearby residuals. **Class I**: A missing absorption line in the model leaves a large, highly correlated patch of negative residuals. **Class II**: An extraneous line in the model leaves a large, highly correlated patch of positive residual. **Class III**: If lines are present in the model but of the wrong strength, many correlated residuals of moderate amplitude will result. The difficulty with class III lines is that for any specific line, there might exist a $\vec{\theta}_\star$ that will fit the line, but there does not exist a $\vec{\theta}_\star$ that will properly fit *all* the lines.

it would be worth discussing by way of a better motivation for using these polynomials. Due to small imperfections in the flux-calibration of the data spectrum, there may be broad regions of the spectrum which are offset from the true continuum level by a slight amount. A traditional approach to address this problem is to normalize both the data and the spectrum to the continuum level before comparison. While this may work well for hotter stars with well-defined continuua, for cooler stars with large molecular features, normalization may introduce artificial features into the spectrum when the pseudo-continuum is incorrectly placed. Instead, we choose to multiply the model spectrum by a low-order Chebyshev polynomial that is normalized to unity. The coefficients of the polynomial, $c_0, c_1, \ldots$, will be free-parameters in our model that we will solve for. If the behavior of the spectrograph is characterized by observations of spectrophotometric standards, then we can put priors on the coefficients that will prevent overfitting of the polynomials. For an example of a typical spectrum, and Chebyshev polynomial, see Figure 1. SA: One thing that has never really been clear to me is how you differentiate the coefficients from the solid angle scaling (or *if* you do that. It would be worthwhile to describe that if necessary especially because I currently see no mention of L_* or R_*/d anywhere in this section (and of course, it should be here somewhere).

With more uncertainty, the polynomial multiplication can also be used as a substitute for flux-calibration since the sensitivity function of an instrument is usually a smooth function with wavelength. When spectra are not flux calibrated, there is greater uncertainty about the properties of the spectrograph and wider priors must be used on the Chebyshev coefficients. In this case, the polynomial might destroy broad-scale information present in the spectrum by fitting it out, something that would be avoided with more accurate priors provided by the flux calibration for the model.

2.3. Likelihood Calculation

SA: I will reserve some comments about notation and symbols for another iteration, but I think you at least want to be very careful about writing the conditional probabilities properly in Eq (5) and (7) – it should always be $p(D|\theta)$ (and note that its θ, not just $\vec{\theta}_\star$). Ultimately, I would like to see a brief synposis first here as well, primarily to describe the nomenclature up front. E.g., what do you mean by D and M? I know what you mean, but you cannot assume the reader does without telling them.

To evaluate which parameters of the model M fit the data set D best we use a standard multidimensional Gaussian likelihood function which allows for covariance between data points. If the vector of residuals is

$$R(\vec{\theta}_\star) = D - M(\vec{\theta}_\star) \tag{3}$$

with a length of N data points, then the likelihood function is

$$p(D|\vec{\theta}_\star) = \frac{1}{\sqrt{(2\pi)^N \det(C)}} \exp\left(-\frac{1}{2} R^T C^{-1} R\right) \tag{4}$$

and the log-likelihood function is

$$\ln[p(\vec{\theta}_\star)] = -\frac{1}{2} R^T C^{-1} R - \frac{1}{2} \ln\det C - \frac{N}{2} \ln 2\pi \tag{5}$$

C is the covariance matrix which describes the covariance between pixels in the spectrum. For independent noise with noise per pixel σ_i, this matrix is diagonal

$$C = \begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ 0 & \sigma_1^2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_N^2 \end{bmatrix} \tag{6}$$

with $\sigma_{ij} = 0$ everywhere and equation 5 reduces to the familiar χ^2 form of a sum over the square of the residuals, weighted by the inverse variance of each data point

$$\ln[p(\vec{\theta}_\star)] \propto -\frac{1}{2}\chi^2 = -\frac{1}{2} \sum_i \frac{R_i^2}{\sigma_i^2} \tag{7}$$

SA: I am not sure how I feel about the dividing line between Sect 2.3 and 2.4. It might make more sense to end right after Eq (5), and then begin Sect 2.4 with a discussion of C and the corresponding likelihood in the sample (independent) case, and then progress into the adopted parameterization. I think I would prefer that, so please wrap up briefly after Eq (5) and push all this text into a re-write of Sect 2.4. Like we discussed, you ought to just write as if your explanation to Tom Loredo is correct; you can always revisit that if need be when you hear back from him.

However, residuals from a spectroscopic fit are often correlated due to
TODO: DUE TO WHAT, exactly?? I'm confused. Wait for response from Tom Loredo. Additionally, systematic errors in the synthetic spectra (such as incorrect line strengths) will result in regions of highly correlated residuals. Properly accounting for correlations in the residuals requires that we use a covariance matrix with off-diagonal terms σ_{ij} and a likelihood function (Equation 5) which uses a matrix product instead of a sum over independent pixels.

We seek to understand and parameterize two types of covariance structure in our model, exemplified in Figure 2: the large scale but generally mild global covariance structure that results from [still to be finalized cause], and a regional covariance structure that is localized to areas where spectral line mismatch results in small patches of highly correlated residuals. Ignoring the complexity of spectroscopic residuals will bias the estimates of the model parameters $\vec{\theta}$.

As a simple analogy, consider the fit of a straight line to a dataset. If the noise in the dataset is correlated, then adjacent data points might be offset from the linear trend in the same direction by a similar amount. If the covariant structure of the noise is ignored, a simple χ^2 will treat these correlated offsets as part of the linear trend, which will result in a biased determination of the slope and intercept of the line, typically with uncertainties that are too small.

2.4. Parameterizing the covariance structure

SA: I am leaving this alone for now, since it obviously needs more work to be edit-able. Give it a shot.

We parameterize the covariance structure using covariance kernels, which sets the covariance between two pixels λ_i and λ_j. This behavior is analogous to the two-point correlation function used in cosmology, where the distance between two galaxies is used instead of wavelength.

Global covariance structure A global covariance structure may result from a slight mismatch in the continuum or mild mismatch of spectral lines like that shown as class 0 in Figure 2. To account for this structure, which is generally mild and typically exists across only a few pixels, we use a *stationary* covariance kernel, which satisfies the property that the degree of correlation only depends on the distance between the two pixels r. Functions like this are also called *radial basis functions*. For spectra, we map r to the velocity

difference between two pixels

$$r(\lambda_i, \lambda_j) = \Delta v = \frac{c}{2}\left|\frac{\lambda_i - \lambda_j}{\lambda_i + \lambda_j}\right| \tag{8}$$

We choose the Matérn $3/2$ kernel because...

TODO: Explore difference between using Matérn kernel and squared exponential kernel. Why Matérn and why Hann? Because they are the vanilla covariance functions that are tried and tested in the machine learning communities.

$$k_{\nu=3/2}(r, a, l) = a\left(1 + \frac{\sqrt{3}r}{l}\right)\exp\left(-\frac{\sqrt{3}r}{l}\right) \tag{9}$$

In order to keep the covariance matrix sparse for computational efficiency, we taper for compact support using a Hann window

$$w(r, r_0) = \frac{1}{2} + \frac{1}{2}\cos\left(\frac{\pi r}{r_0}\right) \tag{10}$$

for a global covariance kernel of

$$k_{\text{global}}(r, a, l, r_0) = w(r, r_0) k_{\nu=3/2}(r, a, l) \tag{11}$$

We add this covariance kernel to the independent variance already present along the matrix diagonal

$$k(\lambda_i, \lambda_j) = k_{\text{global}}(r, a, l, r_0) + \delta_{ij}\sigma_{ij}^2 \tag{12}$$

where δ_{ij} is the Kroenecker delta function.

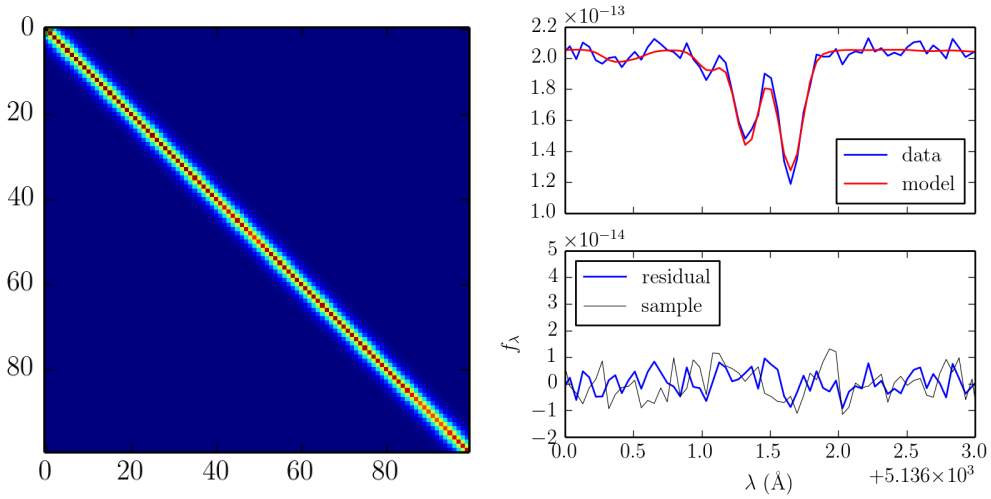

Fig. 3.— **Left** a covariance matrix generated with the Matérn kernel and typical parameters for our dataset. **Right** general spectroscopic residuals near a stellar continuum overlaid with a sample draw using the covariance matrix, showing that the two look similar in structure and amplitude.

TODO: Prettify these figures. Consistent labelling. Change color stretch so Matrix zeros are white.

To visualize what the covariance matrix C might look like when parameterized by Equation 11, see the left panel of Figure 3. To explore the relationship between the parameters of the covariance kernel and the properties of the noise, first consider simulating uncorrelated random Gaussian noise by drawing many independent samples from a Gaussian. This same uncorrelated noise could also be represented as a single draw from a N-dimensional Gaussian with a diagonal covariance matrix, like in Equation 6. If instead of a diagonal covariance matrix, we were to draw samples from a non-trivial covariance matrix like the one in the left panel of Figure 3, then we will see correlated noise like in the right panel.
TODO: Do you think examples like Figure 3 are helpful or a waste of space?

Specific line covariance For the large and highly correlated residuals that result from class I, II, and III line errors, we use a *non-stationary* covariance kernel, which means that the covariance explicitly depends on λ_i and λ_j.
TODO: Explain how we come up with this kernel, which is the covariance resulting from a Gaussian function.

$$k_{\mathrm{G}}(\lambda_i, \lambda_j | a, \lambda_\mu, \sigma) = \frac{a^2}{2\pi\sigma} \exp\left(-\frac{r^2(\lambda_i, \lambda_\mu) + r^2(\lambda_j, \lambda_\mu)}{2\sigma^2}\right) \tag{13}$$

The squared exponential kernel is
TODO: Why do we use the squared exponential and not Matérn?

$$k_{\exp}(r, h) = \exp\left(\frac{-r^2}{2h^2}\right) \tag{14}$$

h is a bandwidth. If h is small, then there will be high-frequency structure. If h is large, then only low-frequency structure will remain. We also use the Hann window to taper the kernel for compact support

$$k(\lambda_i, \lambda_j | h, a, \mu, \sigma, r_0) = w(r, r_0)\, k_{\exp}(r, h)\, k_{\mathrm{G}}(\lambda_i, \lambda_j | a, \mu, \sigma) \tag{15}$$

TODO: Similar paragraph about visualizing the covariance matrix, drawing samples. References to Figure 4.

- non-stationary kernel localizes increased variance to a specific spectral line
- better than just masking out the region, since these contain info
- there will be many sets of regions (∼1-4% of all lines might be covered by a region)
- there can be hyperparameters describing the *population* of regions in a hierarchical model (mostly width, amplitude).

Covariance kernel parameters are parameters in our model that we explicitly sample for.

The benefit mainly comes from simply modelling these systematic residuals to begin with, since a model with covariance is far more likely than forcing the fit, and far more justified than masking regions which do not fit.

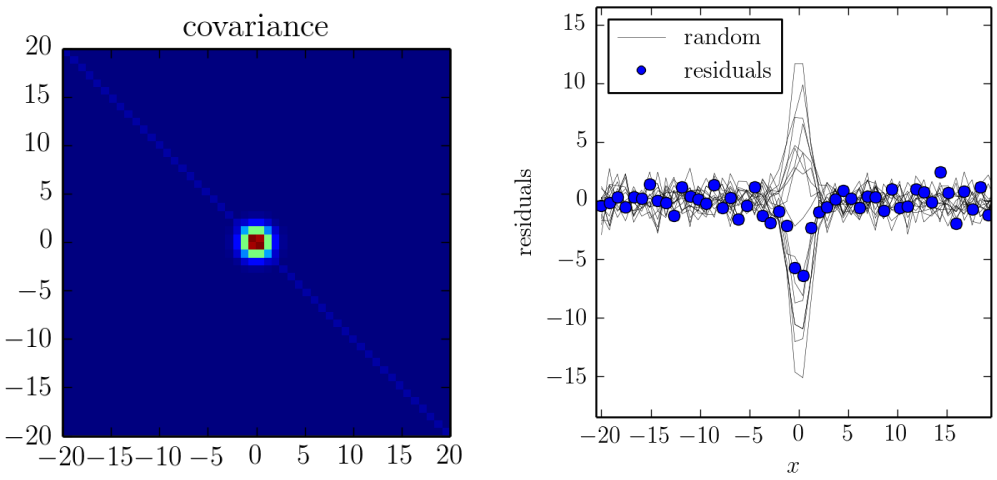

Fig. 4.— **Left** a covariance matrix generated with the region kernel. **Right** general spectroscopic residuals near a mismatched stellar line overlaid with a sample draw using the covariance matrix, showing that the two look similar in structure and amplitude.
TODO: Similar improvements to Figure 3.

TODO: Show how the kernel can accounts for complicated M dwarf structure where there is large, messy mismatch.

2.5. Exploring the posterior

Markov Chain Monte Carlo
TODO: Given this is 2014 and not 2010, how much do you think I need to explain?

2.6. Applications

Learnt covariance structure can be used to correct the models.

3. Tests

TODO: Which tests to show?

This version of the paper was generated from a git repository available at `http://github.com/`

iancze/StellarSpectra/ with git hash d7bcb81 (2014-05-06).

REFERENCES

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

This preprint was prepared with the AAS LaTeX macros v5.0.