Draft version December 28, 2020
Preprint typeset using LaTeX style emulateapj v. 5/2/11

# GSPICE: ARTIFACT DETECTION AND REPAIR IN SPECTRAL DATA

Douglas P. Finkbeiner[1,2,3]

*Draft version December 28, 2020*

## ABSTRACT

We present the GausSian PIxelwise Conditional Estimator (`GSPICE`), a tool for spectral cleaning. GSPICE detects and repairs artifacts in spectral data caused by sensor defects and cosmic-ray hits, and is useful for validation of spectral hardware and data processing systems. Unlike traditional methods that require hardware-specific code, GSPICE takes a data-driven approach, modeling an ensemble of spectra as a multivariate Gaussian and estimating the expected value of each pixel in each spectrum conditional on others. Significant deviation of observed values from these estimates reveals outliers, which can be replaced by their estimates. We provide an implementation of GSPICE in both Python and IDL, and show results based on 3.9 million stellar spectra from the LAMOST survey.

*Subject headings:* methods: data analysis, techniques: spectroscopic, surveys

## 1. INTRODUCTION

Spectral cleaning is an essential part of spectral data analysis. A spectroscopic data set may fall short of perfection in numerous ways. The sensor may have permanent defects like bad CCD columns, or transient defects such as cosmic-ray hits. In a fiber spectrograph, software failures might cause whole sections of a spectrum to be assigned incorrect wavelengths, or cause the spectral trace to be poorly tracked. In the case of hyperspectral imaging, distortions known as *smile* and *keystone* present a challenge. (refs???) Above all, unknown unknowns may lurk in any complex data set. All of these concerns suggest the need for a data-driven reality check, a way to test whether a given observation conforms broadly to expectations.

The question of conformance to expectations is a subtle one. Rare objects may not be represented in training data, but still must be handled properly and not rejected outright. Emission and absorption features from the atmosphere or other material along the line of sight may dominate the variance at some wavelengths. In spite of these issues, we must be able to detect artifacts and repair them with accurate estimates of the mean and variance for each outlier pixel. In particular, correct variance estimates are often important for downstream analysis.

The key insight in this work is that we can think of stellar spectra[4] as being drawn from an $N$-dimensional Gaussian. Each dimension corresponds to a wavelength bin, but these are not statistically independent. The redundancy in stellar spectra is quantified by their covariance matrix $\Sigma$, with element $\Sigma_{ij}$ giving the covariance of wavelength bin $i$ with bin $j$. In general $\Sigma$ is not sparse, and the off-diagonal structure in it connects disparate wavelength ranges for both physical and instrumental reasons. The covariance matrix has a steep eigenspectrum meaning that relatively few dimensions of the spectral vector space explain most of the variance. This is why principal component analysis (PCA) works. Because of the high redundancy, it is possible to estimate any pixel in the spectrum from the others.

There are important limitations to GSPICE. The spectra must be redshifted to $v = 0$ or some other reference velocity. Any spectral absorption features along the line of sight, e.g., diffuse interstellar bands (DIBs), may be substantially Doppler shifted relative to the object of interest creating deviations that do not neatly fit into this framework (but see §3.5). The instrumental line-spread function may vary as a function of wavelength, but must be uniform across the sample.

In Section 2, we present Gaussian conditional estimation as a way to find outlier pixels and estimate the mean and variance of replacement values. We apply GSPICE to the LAMOST data set in Section 3, demonstrating artifact detection and repair, as well as characterization of foreground absorption features. Section 4 summarizes our results and describes future directions. The appendix describes the mathematical and computational methodology that allows us to effectively invert $4000 \times 4000$ matrices in one millisecond, and links to code in Python and IDL.

## 2. THE METHOD

The Gaussian process approach to spectral analysis stands in contrast to the more conventional principal component analysis (PCA; Pearson 1901; Hotelling 1933). In both, we consider a spectrum with $N_\lambda$ wavelength bins to be a point in an $N_\lambda$-dimensional vector space (for LAMOST, $N_\lambda = 3801$). Previous experience (Bolton et al. 2012) has shown that astronomical spectra are not uniformly distributed in this space, but rather occupy a rather low-dimension subspace (10-20 dimensions) of it. This allows dimensionality reduction approaches such as PCA to adequately describe a stellar spectrum with roughly 10 parameters. PCA decomposes the (mean-subtracted) sample into eigenvectors and eigenvalues of the sample covariance matrix, and selects the first $N$ eigenvectors as a basis for an $N$-

[1] Institute for Theory and Computation, Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, MS-51, Cambridge, MA 02138 USA
[2] Department of Physics, Harvard University, Cambridge, MA 02138 USA
[3] Max-Planck-Institut für Astronomie, Königstuhl 17, D-69117 Heidelberg, Germany
[4] For the purposes of this paper we are considering continuum-normalized spectra, but these claims also apply more generally.

dimensional subspace within the larger spectral space.

PCA is a simple and powerful tool, but has some shortcomings. Spectral features may be ignored if they are rare or absent in the training data used to compute the covariance. PCA discards the information about the relative importance of the eignevectors used. The first several are used, and the rest are not. Because of this, the uncertainty of a PCA reconstruction is hard to assess, and is even more challenging in the case of highly variable noise or partially missing data.

### 2.1. *Stellar spectra as a Gaussian process*

A closely related but more sophisticated approach is to assume that spectra are generated by a Gaussian Process, i.e. that appropriately normalized and mean-subtracted spectra are distributed as an $N_\lambda$-dimensional Gaussian in spectral space. This does not have to be true in detail, and of course it is not if the sample is multi-modal. However, this approach retains the full covariance structure of the training sample, allows reconstruction of missing data, incorporates noise estimates, and respects rare objects according to their representation in the training data rather than possibly ignoring them completely.

In the Gaussian process (GP) paradigm, spectra are drawn from an underlying $N_\lambda$-dimensional Gaussian, which is fully specified by a covariance matrix, $\Sigma$.

$$G(x) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp(x^T \Sigma^{-1} x). \tag{1}$$

At this point in the exposition, one sometimes asserts that the GP is *stationary*, meaning that the covariance can be specified by some function of distance between data points (the *kernel*). In this case, the GP can be used to predict data in parts of parameter space – in our case at wavelengths – that have not been observed. This is approximately true for many phenomena, and recognition of the power of this approach, combined with increasingly capable computers and an influential textbook by Rasmussen & Williams (2006), has greatly expanded the use of GPs in machine learning in recent years.

A stationary GP is a powerful technique with wide applicability, but it is not appropriate for spectra. The dimensions of the spectral space correspond to particular wavelengths, and there are physical relationships between these wavelengths. For example, if two spectral bins separated widely in wavelength both contain an oxygen line, the values in those bins are correlated for a physical reason. In this case we have a non-stationary GP with a rich covariance (Figure 1) tying together disparate parts of the spectrum, and that structure is what allows us to estimate one spectral region conditional on another.

### 2.2. *Gaussian Conditional Estimation*

The central goal of this work is to detect and repair bad or missing data in a spectral data set. Specifically, we estimate the values in some spectral pixels[5], $k_*$, conditional on the values in other pixels, $k$. The procedure for Gaussian estimation does not depend on whether the process is stationary or not, merely that we can express the covariance.

[5] We use "pixel" and "wavelength bin" interchangeably in the following.

We represent each spectrum with a column vector, $x$, and the set of values at indices $k$, $x_k$. The matrix $X$ denotes a set of spectra, and the $k$th bin of the $i$th spectrum is $X_{ki}$. The empirical covariance of the reference spectra with themselves, $\Sigma_{kk} = X^T X$, and the covariance of the estimated pixels with the reference pixels $\Sigma_{k_*k}$. Like any covariance matrix, $\Sigma_{kk}$ is a symmetric positive semi-definite matrix, and its inverse can be calculated quickly by Cholesky factorization. (See Appendix A for discussion of computational approaches).

The Gaussian conditional estimate (indicated with a tilde) of the values in pixels $k_*$ of a given spectrum is

$$\tilde{x}_{k_*} = \Sigma_{k_*k} \, (\Sigma_{kk})^{-1} x_k. \tag{2}$$

The matrix $\Sigma_{k_*k} \, (\Sigma_{kk})^{-1}$ may be computed once and applied to an array of spectra, yielding an array of estimates

$$\tilde{X}_{k_*i} = \Sigma_{k_*k} \, (\Sigma_{kk})^{-1} X_{ki} \tag{3}$$

where $X_{ki}$ is the $N_\lambda \times N_{spec}$ array of spectra. The estimated variance is

$$\tilde{\Sigma}_{k_*k_*} = \Sigma_{k_*k_*} - \Sigma_{k_*k} \, (\Sigma_{kk})^{-1} \Sigma_{kk_*} \tag{4}$$

The estimated mean given by Eq. (2) is a linear function of $x_k$, but the variance does not depend on $x_k$. This is a hint that some pre-scaling of the inputs will be necessary, both to have them contribute to the empirical covariance matrix with appropriate weight, and to recover realistic variance estimates. In practice, we scale each spectrum by some measure of $S/N$, estimate, and then scale the mean and variance estimates accordingly.

#### 2.2.1. *Two-dimensional case*

It is difficult to visualize how Equations 3 and 4 play out in 4000 dimensions. To build intuition, consider a toy problem in two dimensions (Figure 2). In this case, we consider a joint probability distribution $P(x, y) \sim \mathcal{N}(\mu, \Sigma)$

$$\Sigma = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_{yy}^2 \end{bmatrix} \tag{5}$$

$$\mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \tag{6}$$

Substituting into Equations 3 and 4 we obtain a conditional distribution $P(x|y) \sim \mathcal{N}(\mu_{x|y}, \Sigma_x)$

$$\mu_{x|y} = \sigma_{xy}^2 \frac{1}{\sigma_{yy}^2} y \tag{7}$$

$$\Sigma_x = \sigma_{xx}^2 - \sigma_{xy}^2 \frac{1}{\sigma_{yy}^2} \sigma_{xy}^2 \tag{8}$$

This is simply the conditional probability, slicing the Gaussian to produce a lower-dimensional Gaussian with mean $\mu_{x|y}$ and covariance $\Sigma_x$.

#### 2.2.2. *Pixelwise estimation*

The Gaussian conditional estimate provides a replacement values for bad pixels based on good pixels – but how do we know which pixels are bad? In practice it can be challenging to know which pixels are corrupted, and evaluation of $\chi^2 = x^T \Sigma^{-1} x$, for $x$ properly scaled, merely

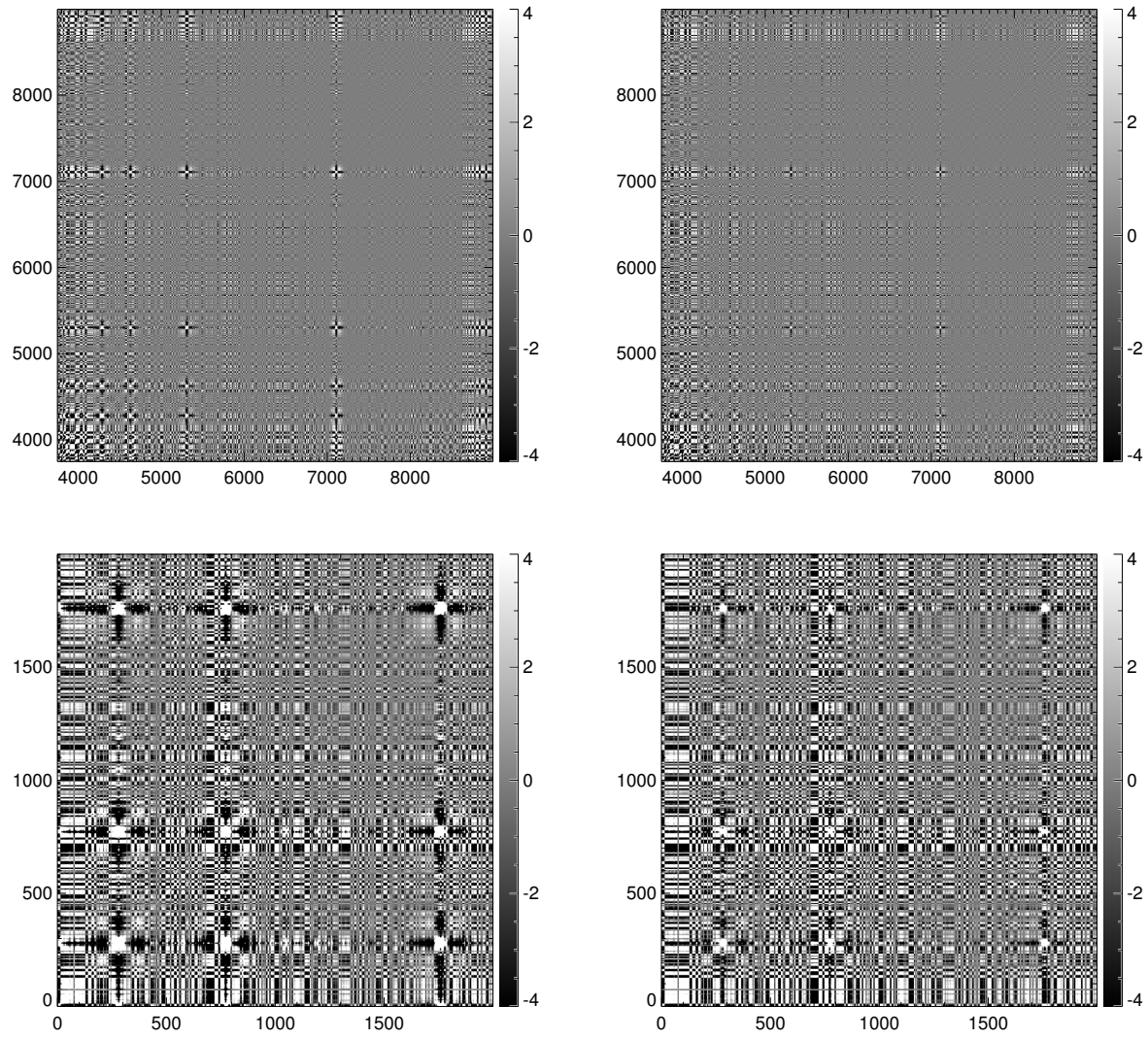


FIG. 1.— Covariance matrix for all stars (left) and low-dust stars (right). The stellar absorption lines (PAH, Na D, etc.) appear, but the most noticeable features are spectral lines arising from the different samples used. Bright spots have a dark cross halo because of the continuum subtraction. Lower panels are the same, but zoomed in.

indicates that a spectrum as a whole is corrupted, but does not tell us what part is bad. This is the primary motivation for GSPICE.

A straightforward approach to identifying corrupted pixels is to loop over pixels in a spectrum and for each one, compute the Gaussian estimate of the mean and variance conditional on all the others. In the case of LAMOST data, the continuum normalization smoothes the spectrum by 20 pixels and then divides by the smoothed spectrum. This imposes a correlation with neighboring pixels within the smoothing window. It is necessary to have a guard window of pixels *not* used for reference. We mask $N_{\rm guard}$ pixels either side of the pixel of interest (Figure 3), with $N_{\rm guard} = 20$ for LAMOST.

There are many conceivable variants of this general framework (e.g., choices of $k_*$ and $k$, and choices made in constructing the covariance, $\Sigma$), but the basic scheme is

- Choose stars with the most unmasked pixels to get compute covariance matrix.
- For each star, estimate masked pixels conditional on unmasked pixels.
- For each pixel, estimate based on others
- compute chi and new mask
- For each star, estimate masked pixels conditional on unmasked pixels using the updated mask.

We now turn to a real-life example.

## 3. GSPICE APPLIED TO LAMOST

In this section we demonstrate the functionality of GSPICE by applying it to data from the Large Sky Area Multi-Object Fiber Spectroscopic Telescope (LAMOST). The Gaussian estimation machinery can be used in two different modes: blockwise and pixelwise.

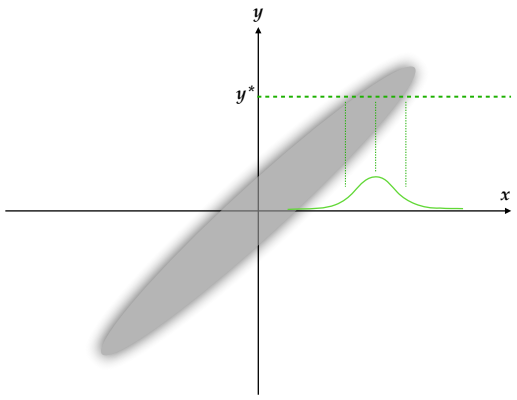


FIG. 2.— Gaussian conditional estimate of $x$ given $y$. A joint distribution $P(x, y)$ is represented by the gray ellipse. The probability density of $x$ conditional on $y = y^*$, $P(x|y = y^*)$, is represented by the solid green line with a mean and variance given by Eqs. (7,8). Equations (3,4) generalize this to provide the probability density of $x_{k_*}$ conditional on $x_k$, $P(x_{k_*}|x_k)$.

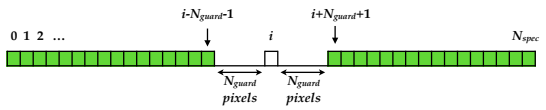


FIG. 3.— Pixels used for Gaussian conditional estimation *(green)*. Pixel $i$ is predicted using all other pixels in the spectrum, except for a guard region around $i$ with radius $N_{\rm guard}$. In this work we take $N_{\rm guard} = 20$, driven by the size of the smoothing kernel used for continuum normalization.

An example of blockwise estimation is estimating the red half of a spectrum from the blue half, or vice versa. This is useful when some problem affects a large wavelength range, such as wavelength miscalibration. A drawback is that because only half of the data is used as a basis for prediction, the prediction is somewhat noisier.

In the pixelwise case, small artifacts (bad columns, CR hits) can be detected with greater sensitivity. To find subtle defects it is helpful to use as much spectral range as possible as the basis for prediction for each test pixel, and then loop over test pixels. This requires more computation, but is often tractable.

Blockwise and pixelwise estimation can be done per star, respecting a pre-existing bad-pixel mask, or they can be applied to all stars at once.

### 3.1. *LAMOST data*

The LAMOST telescope has an effective aperture of 4m with a 5° field of view. The instrument is comprised of 16 spectrographs fed by 250 fibers each, for a total of 4000 fibers with 3.3 arcsecond diameter. In each spectrograph, light passes through a dichroic element to a red camera and a blue camera, each with a 4k×4k CCD. The red and blue sides of the spectrum are combined into a composite spectrum with wavelength bins uniform in $\log\lambda$. The bin width corresponds to $\Delta\log\lambda = 10^{-4}$, or about 69 km/s. The fifth data release[6] contains 9,026,365 spectra with a limiting magnitude of $r = 19$ at resolution 1800, covering a wavelength range from 3750-9000 Å.

We use a sample of 3.9 million spectra with $S/N > 30$ in i-band. There are numerous quality flags such as ???. In some cases we ignore the quality flags to demonstrate what would happen in the case of unexpected artifacts.

### 3.2. *Blockwise GCE for Large-scale artifacts*

Blockwise use of GCE is convenient for finding problems that affect a broad wavelength range, but are assumed to be isolated from some other wavelength range. In the following example, we apply Eqs. (3) and (4) to the LAMOST sample, with the pixels to estimate ($k_*$) corresponding to the entire red half of the spectrum, conditional on $k$, the blue half of the spectrum. In roughly 99% of cases, the prediction agrees broadly with the observation, but in some cases there are striking differences. Figure 4 shows a set of spectra from one field, ordered by spectrograph. These composite spectra contain data from more than one exposure, and it appears that in at least one exposure, some of the spectrographs have an incorrect wavelength calibration.

### 3.3. *Pixelwise GCE: Finding outliers*

Pixelwise GCE (or GSPICE) aims to provide a more sensitive detection of localized outliers by using all information in the spectrum except that near the test pixel. that is, $k_*$ is a single pixel and $k$ is all other pixels except those in a guard window of radius $N_{guard}$ (Figure 3). This is slower than blockwise GCE, because we must loop over each pixel.

The thought of looping over $\sim 4000$ pixels in $\sim 4$ million spectra, requiring 16 billion effective matrix inversions, may be somewhat daunting. Even if Eqs. (3) and (4) can be evaluated in 1 ms, this is still 16 Ms or about 4000 core hours. For maximum sensitivity to outliers this can be done. However, the following shortcut is effective.

We propose to

- Compute the covariance matrix
- for each star, replace masked pixels with estimate ($N_*$ inversions)
- for each pixel, estimate that pixel for all stars ($N_\lambda$ inversions)
- compute $\chi$, mask, replace with estimate.

[6] Available at http://dr5.lamost.org

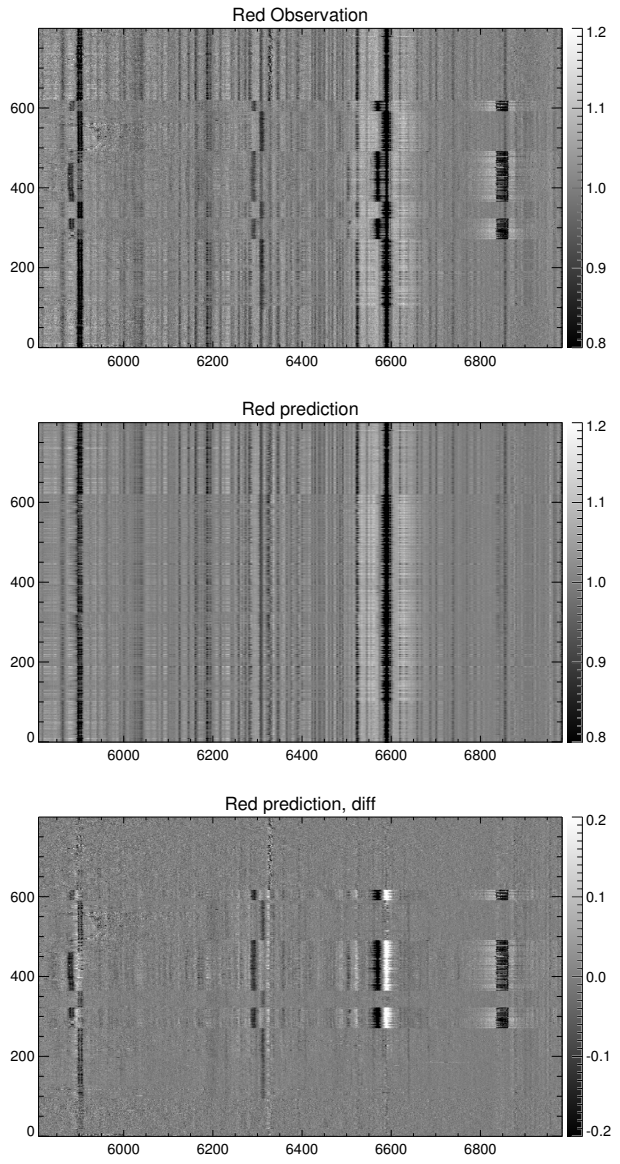


FIG. 4.— Block-wise GSPICE, predicting the red side of the spectrum from the blue side. Some observed spectra (*upper*) contain lines shifted relative to the GSPICE prediction (*middle*), especially apparent in the difference image (*lower*). GSPICE predicts mean and variance, so non-conforming pixels may be found with a simple $\chi^2$ cut. This wavelength problem does not appear in LAMOST DR4, but affects $\sim 1\%$ of the spectra in DR5.

- iterate
- using $5/10\,\sigma$ cut, go back and for each star estimate bad pixels. ($N_*$ inversions)

This requires $2N_* + 2N_\lambda$ matrix inversions, enormously faster than the alternative. The result of this procedure may be seen in Figure 5.

### 3.4. *Artifact repair*

The previous sections were focused on artifact detection. As a final step, we can replace the outlier pixel mean and variance values with the GSPICE estimates. This requires a criterion for determining a pixel is bad enough to replace. There is a natural tension between replacing too few (leaving artifacts that could easily have been repaired) and too many (possibly destroying information, e.g., about rare spectral features and unusual

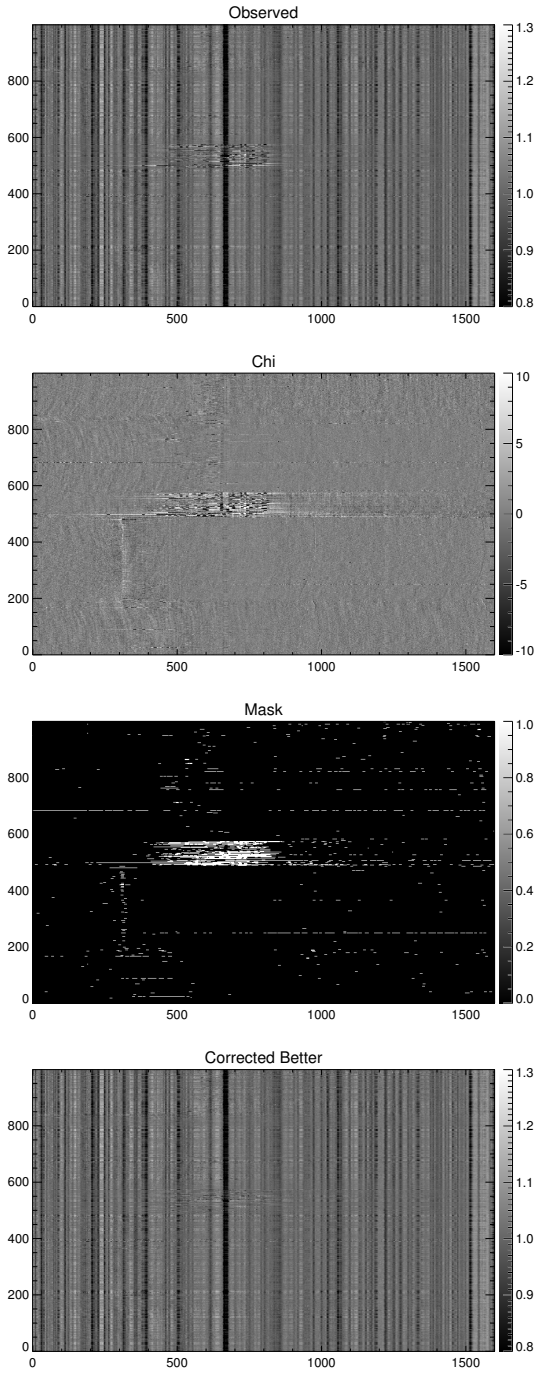


FIG. 5.— Repair of artifacts in LAMOST data. Each row corresponds to a spectrum over the wavelength range ??? to ???. GSPICE is applied to the observed data (*upper left*) to estimate the mean and variance. These are used to evaluate $\chi$ =(Observed-Predicted)/$\sqrt{\text{var}}$ (*upper right*). Pixels with $|\chi| > 10$ are masked, and the mask is dilated by 2 pixels in the wavelength direction (*lower left*). Finally, masked pixels are replaced with the GSPICE prediction (*lower right*). The subtle arcs in the $\chi$ image are below threshold for masking, but may reveal something interesting about the data processing.

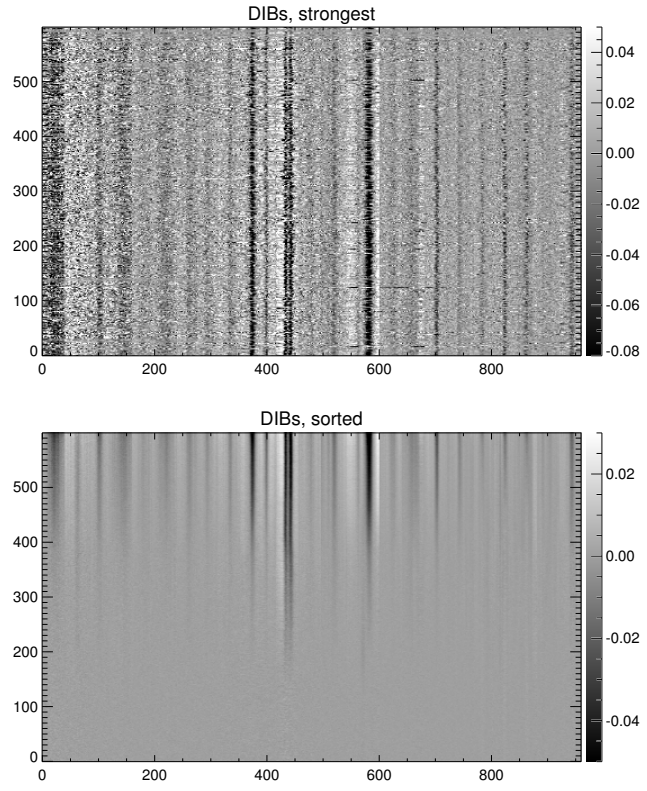


FIG. 6.— 150,000 spectral (averaging 500 per row) sorted by foreground dust.

metal abundances). We adopt the convention that $10\sigma$ outliers are replaced, and their estimate is conditioned on pixels $< 5\sigma$ out. In other words, pixels deviating by more than $5\sigma$ but less than $10\sigma$ are not bad enough to discard, but not good enough to use in the estimate. In Figure 5 the repaired image is constructed in this way.

The GSPICE code delivers

- original spectrum
- original variance
- original mask
- GSPICE predicted mean
- GSPICE predicted variance
- repaired spectrum
- mask

### 3.5. *Line-of-sight absorption features*

To demonstrate the use of GSPICE for anomaly detection, we make use of diffuse interstellar bands (DIBs) and atomic absorption features. DIBs are absorption features with varying widths and strengths, some well correlated with reddening, some not. Though they were recognized decades ago (Merrill 1934; Herbig 1975), and over 400 have been cataloged (Hobbs et al. 2009), very few have been conclusively identified. They are thought to arise from CH stretching modes in small carbonaceous dust grains, and are worthy of study in their own right as a tracer of interstellar chemistry.

Measuring ISM absorption of background starlight depends on an accurate estimate of the unabsorbed stellar spectrum. This is especially challenging for the Na D doublet, which appears prominently in a range of stellar types. High resolution spectroscopy can separate the much narrower ISM doublet from the stellar doublet, but for more modest resolutions (e.g. SDSS, LAMOST, DESI), instrumental broadening prevents using the line width as a distinguishing feature. Fortunately, estimation of the unabsorbed spectrum near features of interest, conditional on the rest of the spectrum, is precisely what GSPICE provides.

It is necessary to build both a "clean" covariance matrix and a "dusty" one. The clean covariance is computed from stars with very low foreground reddening ($E_{B-V} < 0.04$). This covariance is used to establish the unabsorbed spectrum. An estimate based on the clean covariance does not "know" about ISM absorption, so it cannot use weak unmasked DIBs to corrupt the estimate. However, such a covariance would be inappropriate for outlier identification because it would flag genuine absorption features as outliers. We therefore build a dusty covariance matrix with no cut on reddening.[7] The variance of pixels is higher near these features, and they covary appropriately. After a pass of GSPICE to flag outliers, we use the clean covariance to predict each stellar spectrum in the regions of interest. Subtracting the predicted spectrum from the observation yields the measurement.

We select wavelength regions within 10 pixels of the 20 strongest DIBs in the LAMOST spectral range from the Hobbs et al. (2009) catalog, and also include the Na D doublet ($\lambda 5890, 5896$) and the KI $\lambda 7666$ and $\lambda 7700$ lines. As a measure of foreground absorption, we sample from the Bayestar 19 3-D dust map (Green et al. 2019) at distances drawn from the *Gaia* parallax (Gaia Collaboration et al. 2018; Luri et al. 2018), using 100 samples to determine the value and variance. Most of the absorption spectra for the 300 stars with highest foreground dust exhibit absorption features (Figure 6).

It is evident that there is some non-linearity. For selected DIBs and Na D, the dependence on reddening is linear at low column and then begins to saturate (Figure 7).

The motivation for this work was not to measure DIBs, but this is perhaps a novel approach. LAMOST resolution of $\sim 150$ km/s (with 69 km/s pixels) is not ideal for getting dust radial velocities as in Tchernyshyov et al. (2018), but this approach could be valuable for such work, or any other time that an unabsorbed continuum is needed. In the same vein, Na D estimates for the *stars* are probably also more reliable with this technique, though the estimates undoubtedly leverage the presence of other metals in the spectrum. Alternately one could rely on Bayestar19 and a sufficiently sophisticated model of interstellar absorption to make the correction. Both approaches could be applied to large spectroscopic samples, but rely on assumptions that must be carefully

[7] For the moment we neglect the fact that these two estimates of the covariance may be constructed with rather different stellar types. We will return to this later.

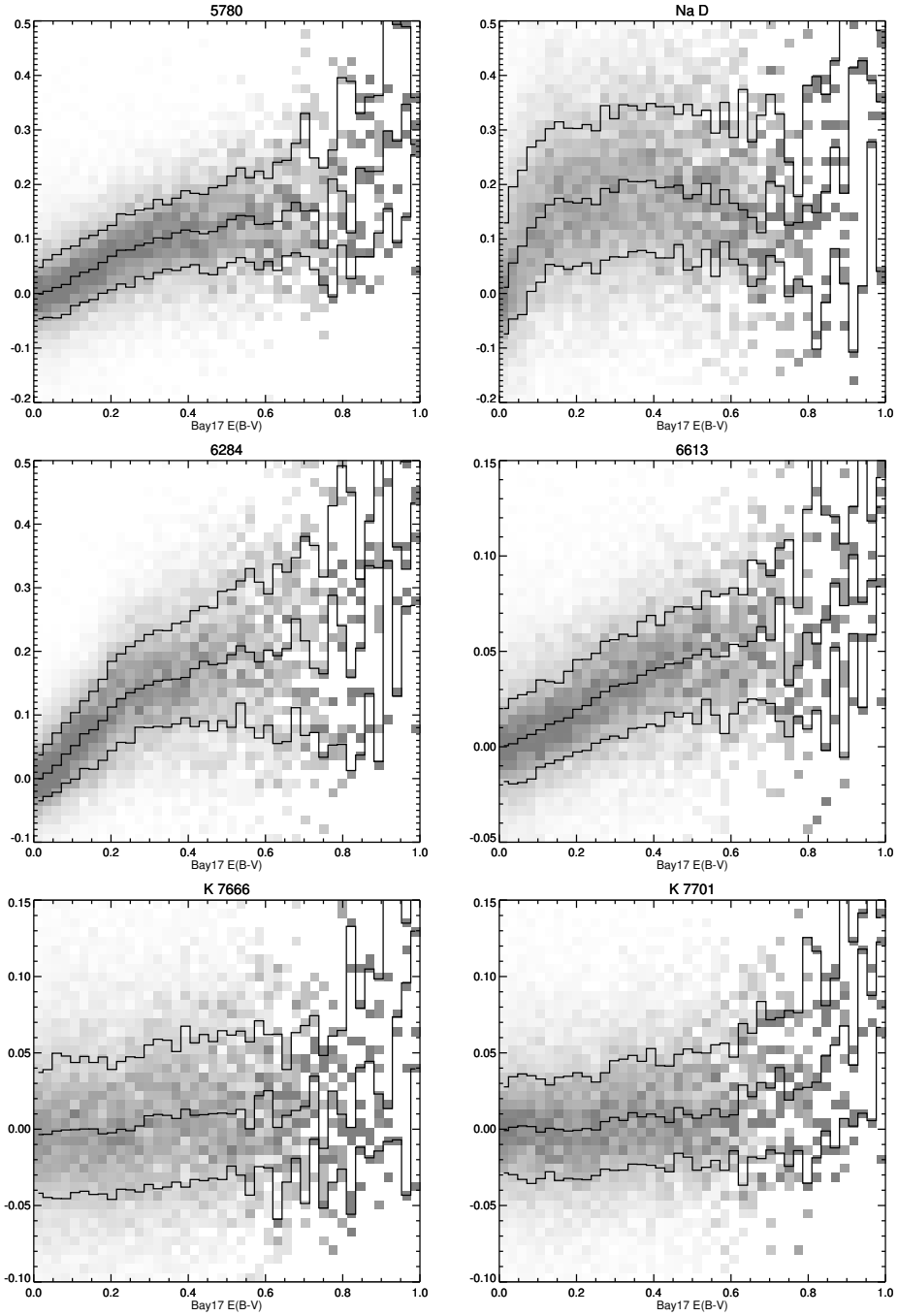


FIG. 7.— Absorption line strength vs. Bayestar19 dust for selected DIBs and atomic lines.

checked with high-resolution spectroscopy.

### 3.6. *Assessing results vs. stellar type*

We aspire to look at reconstruction quality vs. $T_{eff}$, $\log g$, Fe/H, etc.

## 4. SUMMARY AND FUTURE DIRECTIONS

Motivated by the need to find outliers in spectral data, we have applied the concept of Gaussian conditional estimation in novel ways. By thinking of stellar spectra as generated by a (non-stationary) Gaussian process, we can ask if a particular spectrum conforms to expectations embodied in a covariance matrix. Astrophysical deviations as well as those caused by instrumental defects and software bugs can readily be detected. Importantly, unknown unknowns may be revealed, because the expectation is not crafted with expert knowledge, but rather specified by a training data set. This data-driven approach has risks and benefits in common with many other ML approaches.

The novelty here is the pixelwise nature of the estimate, looping over pixels and predicting each conditional on the rest of the spectrum. This approach finds localized defects with impressive sensitivity. If some spectral regions are known to be suspect (e.g near ISM absorp-

tion features) they can be predicted conditional on the rest of the spectrum.

Some generalizations beyond the scope of this work:

- Can we measure radial velocity by preparing many covariance matrices corresponding to different velocities shifts, then evaluating $\chi^2$ for each?
- Can we apply this technique to continuum-subtracted spectra rather than continuum-normalized? Variations in sensitivity across the spectral range would be represented in the training data and effectively ignored.
- Can the covariance matrix be supplemented with ancillary data (colors, magnitudes, reddening estimates, abundances) and then used to predict those quantities? The methodology may seem too linear for that, but one could input a combination of data, data$^2$, data$^3$, ...

This last is probably better done with an inherently non-linear approach (neural networks) but for artifact detection and repair, GSPICE is easy to understand and computationally straightforward.

*Acknowledgments:* Guoshoujing Telescope (the Large Sky Area Multi-Object Fiber Spectroscopic Telescope LAMOST) is a National Major Scientific Project built by the Chinese Academy of Sciences. Funding for the project has been provided by the National Development and Reform Commission. LAMOST is operated and managed by the National Astronomical Observatories, Chinese Academy of Sciences. This work has made use of results from the European Space Agency (ESA) space mission *Gaia*, the data from which were processed by the *Gaia Data Processing and Analysis Consortium* (DPAC). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the Gaia Multilateral Agreement. The *Gaia* mission website is https://www.cosmos.esa.int/web/gaia.

REFERENCES

Bolton, A. S., Schlegel, D. J., Aubourg, É., et al. 2012, AJ, 144, 144
Gaia Collaboration, Brown, A. G. A., Vallenari, A., et al. 2018, A&A, 616, A1
Green, G. M., Schlafly, E. F., Zucker, C., Speagle, J. S., & Finkbeiner, D. P. 2019, arXiv e-prints, arXiv:1905.02734
Herbig, G. H. 1975, ApJ, 196, 129
Hobbs, L. M., York, D. G., Thorburn, J. A., et al. 2009, ApJ, 705, 32
Hotelling, H. 1933, Journal of Educational Psychology, 24, 417
Luri, X., Brown, A. G. A., Sarro, L. M., et al. 2018, A&A, 616, A9
Merrill, P. W. 1934, PASP, 46, 206
Pearson, K. 1901, Philosophical Magazine, 2, 559
Rasmussen, C., & Williams, C. 2006, Gaussian Processes for Machine Learning, Adaptive Computation and Machine Learning (Cambridge, MA, USA: MIT Press), 248
Tchernyshyov, K., Peek, J. E. G., & Zasowski, G. 2018, AJ, 156, 248

## APPENDIX

## COMPUTATIONAL DETAILS

This work used the Cannon[8] cluster at Harvard University. Each Cannon node has two water-cooled Intel 24-core Platinum 8268 Cascade Lake processors with 4 GB of RAM per core. Cascade Lake cores have dual AVX-512 fused multiply-add (FMA) units[9] For matrix multiplication we use CBLAS dgemm routines[10] from Intel's Math Kernel Library in IDL, and the MKL version of NumPy in Python. These both achieve about 80 GFlops or 70% of the theoretical maximum for matrices of the size used in this work. On a 48-core node that amounts to 3.8 TFlops (double precision).

However important matrix multiplication may be for GSPICE, matrix *inversion* would dominate if not approached thoughtfully. A covariance matrix is positive semi-definite and symmetric, allowing use of fast Cholesky routines. Even using Cholesky decomposition, it takes roughly 1 core-second to invert a $3801 \times 3801$ matrix with `dpotri` and `dpotrf`[11]. Outlier finding involves inverting 3801 matrices and applying each one to $N$ spectra. Repair involves inverting one matrix per spectrum, because each spectrum (in general) has a different set of pixels masked. This requires millions of matrix inversions for a survey like LAMOST, and if rejection is iterated, perhaps billions.

In this section we use the fact that we are inverting many matrices that are almost identical to achieve a speedup of 2-3 orders of magnitude. We have the full $N \times N$ covariance matrix, $M$, and repeatedly extract a submatrix $A$ containing most of the rows and columns of $M$. In the following we present a way to compute $A^{-1}$ given that we already have $M^{-1}$.

### *Inversion of a submatrix*

The first step is to relate the inverse of a submatrix of $M$ to $M^{-1}$. Suppose $A$, $B$, $B^T$, and $D$ are $p \times p$, $p \times q$, $q \times p$, and $q \times q$ matrices[12], and $D$ is invertible. Let

$$M = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \tag{A1}$$

so that $M$ is a $(p+q) \times (p+q)$ matrix. It's inverse has blocks

$$M^{-1} = \begin{bmatrix} P & Q \\ Q^T & U \end{bmatrix} \tag{A2}$$

By the matrix inversion lemma,[13] the inverse of $A$ is

$$A^{-1} = P - QU^{-1}Q^T \tag{A3}$$

To verify, multiply both sides of Eq. (A3) by $A$.

$$A^{-1}A = PA - QU^{-1}Q^TA \tag{A4}$$

Because $M^{-1}M = I$,

$$Q^TA + UB^T = 0 \tag{A5}$$

$$PA + QB^T = I \tag{A6}$$

Combining these with Eq. (A4) verifies the identity, as long as $U$ is invertible. While Eq. (A3) is faster to evaluate than a direct inverse (See Figure 8) it is usually advantageous to set up the problem so that $A^{-1}$ is never evaluated.

### *Inverse of a submatrix times a vector*

In this section we consider the product of $A^{-1}$ and a vector $y$, and mention a few tricks to expedite the calculation. Building on Eq. (A3) we have

$$A^{-1}y = Py - QU^{-1}Q^Ty \tag{A7}$$

Let's also assume we are going to apply this equation many times, so it is practical to pre-compute $M^{-1}$ and $M^{-1}y$. Let $Y$ be the full vector of length $p+q$, and $y = Y[k]$, where $k$ is the index list of pixels to keep, and $y_r = Y[r]$, with $r$ the list of pixels to remove. Note that

$$Q^Ty = (M^{-1}Y)[r] - Uy_r \tag{A8}$$

[8] Named in honor of Annie Jump Cannon, who developed the first spectral classification system for stars.

[9] AVX-512 is the 512-bit Advanced Vector Extensions Single Instruction Multiple Data (SIMD) instruction set for x86 cores. Registers can hold 512 bits of data, for example 8 double-precision floats. Fused Multiply Add means that the numbers in a register may be multiplied by those in another, and added to those in a third, all on one clock cycle. With 2 FMA units, a core can perform $8 \times 2 \times 2 = 32$ double-precision operations per cycle. A clock speed of 3.5 GHz implies a theoretical maximum of 112 billion floating-point operations (Gflops) per core.

[10] MKL C Basic Linear Algebra Subprograms (CBLAS) dgemm

[11] LAPACK dpotri

[12] row-major notation, so "row" × "column".

[13] This is the Schur complement of block $U$ of $M^{-1}$.

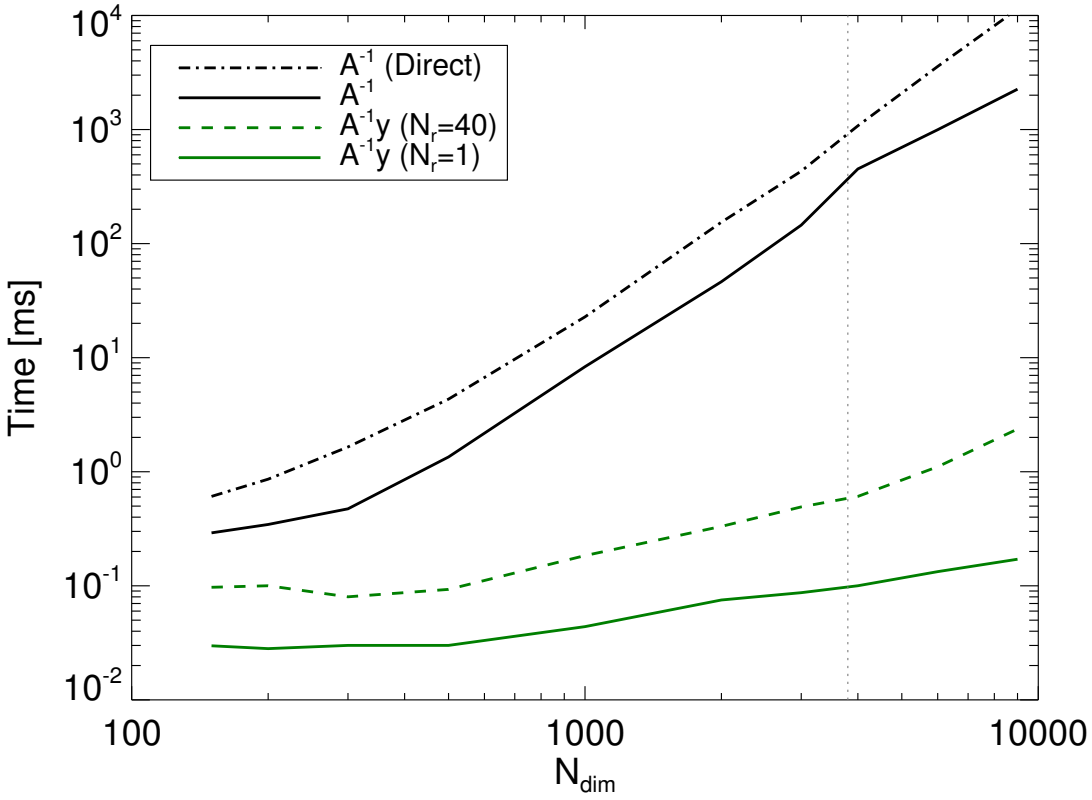


FIG. 8.— Time required for inversion of submatrix $A$ of $M$, where $M$ is $N_{dim} \times N_{dim}$. $A$ has one row and column removed. Direct inversion of $A$ via Cholesky factorization is several times slower than the expression in Eq. (A3). Computation of $A^{-1}y$ is faster still via Eq. (A7). The vertical line at $N_{dim} = 3801$ shows the case relevant for LAMOST data. All calculations are float64.

The right hand side evaluates much faster than $Q^T y$ for $N_r \ll N_k$.

Next we need $U^{-1}Q^T y$, but $U^{-1}$ is positive semi-definite, so we compute the product using Cholesky backsubstitution instead of directly inverting $U$. $U^{-1}Q^T y$ is a column vector of length $N_r$ and directly multiply $Q$ by it.

These steps are fast enough that $Py$ would dominate the execution time, but $Py = (M^{-1}y)[k] - Qy[r]$. Using all of these shortcuts, the time required to evaluate $A^{-1}y$ for $N_k \sim 4000$ and $N_r \sim 40$ is 3 orders of magnitude faster than a direct inversion of $A$ (Figure 9).

Most use cases of GSPICE involve either prediction of a single pixel for many spectra, or prediction of many pixels in a single spectrum. This amounts to either $y^T A$ or $Ay$. In other cases, many pixels for many spectra may be needed, but in this case matrix inversion generally doesn't dominate the time budget. So for timing estimates at $N_\lambda = 4000$, we can roughly say that inversions are 1ms and matrix multiplication consumes the rest of the time.

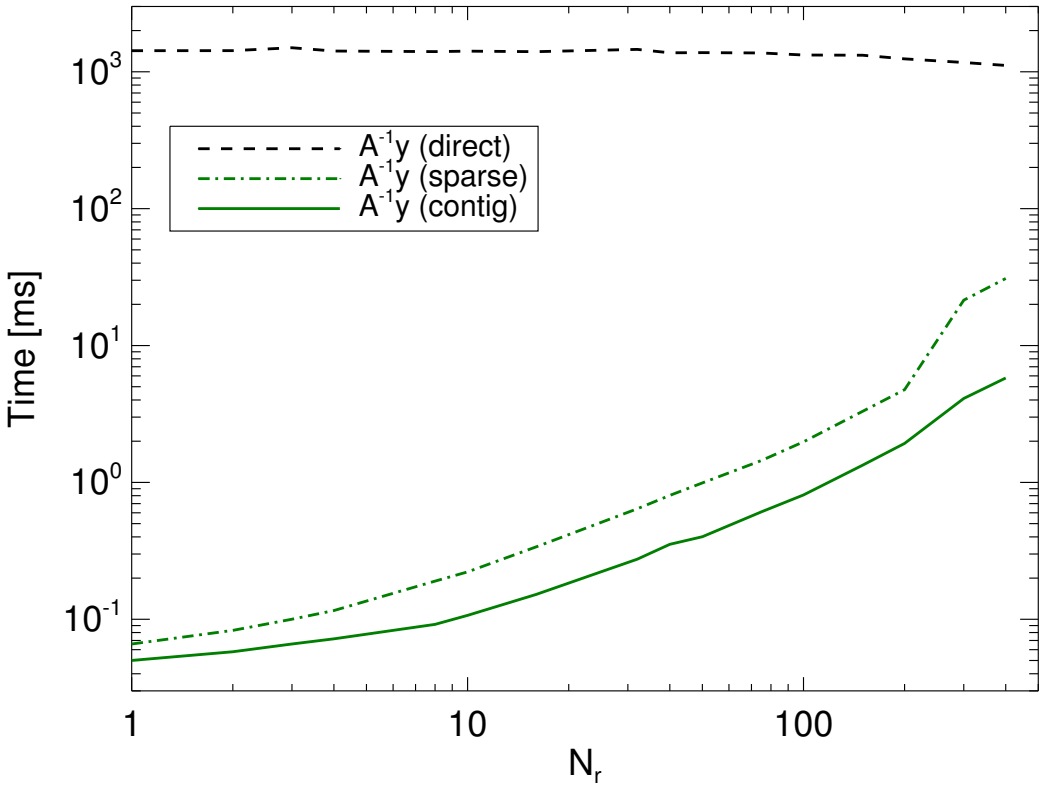


FIG. 9.— Time required for computations involving a submatrix of a $4000 \times 4000$ matrix, $M$, and a column vector, $y$. In all cases $A$ is a submatrix of $M$ with $N_r$ rows and columns removed. Time to calculate $A^{-1}y$ with Cholesky factorization of $A$ to invert and then multiplying by $y$ is nearly constant as a function of $N_r$ (direct). Using Equation A7 for $A^{-1}y$ is 3 orders of magnitude faster for $N_r = 40$ in general (sparse) and even faster if the removed rows and columns are contiguous (contig). Execution time is approximately $(8N_r + 40)\mu$s. All calculations are float64.

## SOFTWARE

In this section we provide pointers to software and examples.