RHMF for astronomy

David W. Hogg, Thomas Hilder

September 17, 2025

1 Introduction

Prior work:

- At the beginning of time, there was principal components analysis (PCA) [**?**], the last universal common ancestor (LUCA) of all self-supervised (or unsupervised) machine-learning methods. PCA replaces (or models) the $N \times M$ rectangular data Y with a low-rank matrix L that minimizes the sum of squares of the residual $Y - L$, subject to $\mathrm{rank}(L) = K < \min(N, M)$. Like almost all of its machine-learning-method descendants, PCA requires complete, rectangular data; it treats every data point identically. It is extremely sensitive to outliers; a single bad pixel in one data record can spoil many or all of the delivered eigenvectors. After all, it is a model to minimize unmodeled *variance* (squared error); whereas the empirical variance in a block of data can easily be dominated by one or a few pixels.

- Unrelated to the successes of PCA, humankind evolved, looked at the stars, embraced weighted least squares (chi-squared fitting) [**?**], got upset about the influences of outliers, and immediately started sigma clipping [**?**]. This has been the dominant method for making weighted-least-squares fits insensitive to rare outliers or data anomalies; that is, to make them more *robust*. [HOGG: Algorithm??] This method is subject to a kind of "mode collapse" in which large amounts of data get clipped out and the model just doesn't represent those data at all.

- If ancient astronomers had looked at the statistics literature, they would have seen iteratively reweighted least squares (IRLS) [**?**]. This method is really a family of methods; but there is a form you can write down [HOGG DO THAT HERE] in which IRLS has all the good properties of weighted least squares with sigma-clipping (and a similar kind of anomaly threshold parameter), but is way less prone to mode collapse. The IRLS method is a workhorse in many domains; it has been used in astronomy now and then [**?**].

- A mathematically rigorous but robust empirical model for (representation for?) data is the robust principal components analysis (Robust-PCA) method of Candés et al [1]. This method attempts to describe the rectangular block of data Y as a sum of a low-rank block L and a sparse block S. Interestingly, the method attempts to make this exact, such that $Y = L + S$ exactly. When the Robust-PCA algorithm is iterated for finite time, it comes finitely close [HOGG: CORRECT?]. Fundamentally, the Robust-PCA is an alternation of a singular value decomposition (with a threshold on the singular values) to

make L and an outlier identification (with a threshold on the residual) to make S.

- Along a separate thread, Tsalmantza & Hogg introduced heteroskedastic matrix factorization (HMF) [2] as a data-driven dimensionality reduction for astronomical spectra (or other kinds of noisy data). The HMF model of rectangular data Y is the rectangular matrix L of rank $K < \min(N, M)$ that minimizes the chi-squared residual (weighted sum of squares). HMF is a replacement for PCA that does not require complete data (because missing data can be assigned vanishing weights), and it has the satisfying property that every data point is weighted with its associated inverse uncertainty variance, which represents the amount of information it brings. However, HMF is still very sensitive to outliers or anomalies in the data, if they are not weighted appropriately.

The algorithm that follows—Robust-HMF—is very much a mash-up of HMF [2] and Robust-PCA [1]. It adds to Robust PCA data weighting and the ability to handle missing and low-information pixels. It sacrifices the nice property of Robust-PCA that it explicitly decomposes the data (or the model for the data) into sparse and low-rank components. It also [HOGG THINKS] can't be expressed precisely as the optimization of a single scalar objective function, which makes it harder to analyze mathematically. [TOM: I think we can express things either as a heavy-tailed MLE, or as a latent-variance hierarchical model with a Gaussian likelihood. See below and Appendix. The latter is undercooked, I only started thinking about it this morning, but I think it's ultimately the nicest. Either way, we have nice statistics.] However, empirically, it works very well in standard astronomical contexts, as we will see.

2 Assumptions and choices

3 Method

The setup is that there are N spectra y_i ($1 \leq i \leq N$), each of which has M pixels y_{ij} ($1 \leq j \leq M$). Importantly for what follows, some of the pixels can have dummy or null values, because not all pixels of all spectra will have been observed in most real contexts. Along with each pixel value y_{ij} there is an associated inverse-variance weight w_{ij}. Usually these weights will be inverse uncertainty variances σ_{ij}^{-2}. Importantly, the pixels with dummy or null values of y_{ij} should have vanishing weights w_{ij}. Vanishing weights ($w_{ij} \to 0$) are equivalent to infinitely large "error bars" ($\sigma_{ij} \to \infty$).

The weights need to have two properties for the method to be safe: *(1)* All the weights need to be strictly non-negative, and *(2)* each weight w_{ij} needs to have units that are inverse of the square of the units of the pixel value y_{ij}. So if the y_{ij} have units of flux, then the units of the weights w_{ij} need to be inverse flux squared.

The data y_{ij} are rectangular in the following sense: Each index value i corresponds to one, well-defined spectrum of one star, and each index value j corresponds to one, well-defined wavelength λ_j (which could be spectrograph-frame wavelength or stellar rest-frame wavelength, depending on task). Every i, j pair has an entry, although many of them will have $w_{ij} = 0$ because they are unobserved values. Because the method will be perfectly insensitive to pixels with $w_{ij} = 0$, it doesn't matter what the fill value is for y_{ij}, but it makes sense to use something sensible (like zero) for numerical safety.

The method will be unstable unless, for each star i, there are many observed wavelengths j ($w_{ij} > 0$ for many pixels j, for every i). Similarly, we will need it to be the case that, for each wavelength j, there are many observed stars i ($w_{ij} > 0$ for many stars i, at every j). Without breaking rectangularity, any stars i or wavelengths j that do not meet these criteria can be dropped from the method prior to start.

The model, conceptually, is that the rectangular data matrix composed of the y_{ij} can be represented, up to noise, as a low-rank matrix plus sparse outliers. To be slightly more specific,

$$y_{ij} = \sum_{k=1}^{K} a_{ik}\, g_{kj} + \text{outliers} + \text{noise} \;, \tag{1}$$

where the rank is K, each a_i is a K-vector of coefficients a_{ik}, each g_k is an M-vector eigenvector of the low-rank model, and the outliers and noise can't really be distinguished from one another. That is, the model is low-rank plus outliers plus noise but really it can be seen as just low-rank plus residuals.

HOGG: probabilistic interpretation of chi-squared, and then probabilistic interpretation of IRLS in this context; is there one? TOM: Yes, see below, especially final section of Appendix A where I mention some generalisation stuff. If the part about hierarchical latent variances is unclear let me know and I can expand it.

Training: Training of this model proceeds by alternating least squares with IRLS mixed in. Before starting the optimization, the coefficients a_{ik} and components g_{kj} are initialized with a singular-value decomposition (SVD) of the data:

$$[Y]_{ij} = y_{ij} \tag{2}$$

$$U\, S\, V = \text{svd}(Y) \tag{3}$$

$$a_{ik} \leftarrow [U]_{ik}\, [S]_{kk} \tag{4}$$

$$g_{ik} \leftarrow [V]_{kj} \;, \tag{5}$$

where Y is the rectangular data matrix, and the operator svd() takes the standard singular-value decomposition with U, V unitary and S diagonal. This initialization is not great, because the data are filled with missing values and the SVD has no understanding of that.

After initialization, the steps in the optimization schedule are the following: The **a-step** finds best-fit values for the coefficients a_{ij} given the current guess of the eigenvector components g_{kj}:

$$a_i \leftarrow \text{solve}(A_i, b_i) \tag{6}$$

$$[A_i]_{kk'} = \sum_{j=1}^{M} g_{kj}\, w_{ij}\, g_{k'j} \tag{7}$$

$$[b_i]_k = \sum_{j=1}^{M} g_{kj}\, w_{ij}\, y_{ij} \;, \tag{8}$$

where a_i is a K-vector of coefficients, the operator solve(A, b) returns $A^{-1}\, b$, A_i is a $K \times K$ matrix with entries $[A_j]_{kk'}$, and b_i is a K-vector with components $[b_i]_k$.

The **g-step** finds best-fit values for the eigenvector components g_{kj}

$$g_j \leftarrow \mathrm{solve}(A_j, b_j) \tag{9}$$

$$[A_j]_{kk'} = \sum_{i=1}^{N} a_{ik}\, w_{ij}\, a_{ik'} \tag{10}$$

$$[b_j]_k = \sum_{i=1}^{N} a_{ik}\, w_{ij}\, y_{ij} \ , \tag{11}$$

where g_j is a K-vector of eigenvector components, A_j is a $K \times K$ matrix with entries $[A_j]_{kk'}$, and b_j is a K-vector with components $[b_j]_k$.

This model has a huge set of degeneracies: The a_{ik} can be multiplied by a factor, and the g_{kj} can be divided by that same factor. The orientation of the eigenvectors g_k can be rotated and the coefficients a_{ik} can be de-rotated. In practice, to break some (but not all) of these degeneracies, after each iteration of the a-step and the g-step, an SVD is performed on the low-rank matrix to rotate and rescale the g_k axes back to a standard orientation in the data space. This looks like the following:

$$[L]_{ij} = \sum_{k=1}^{K} a_{ik}\, g_{kj} \tag{12}$$

$$U\, S\, V = \mathrm{svd}(L) \tag{13}$$

$$a_{ik} \leftarrow [U]_{ik}\, [S]_{kk} \tag{14}$$

$$g_{ik} \leftarrow [V]_{kj} \ , \tag{15}$$

similar to the SVD-based initialization described above, but now acting on the low-rank model rather than the data. Importantly this operation does not change the model predictions *at all*; it just re-scales, re-orients, and re-orders the eigenvectors.

The **w-step** updates the weights w_{ij} for robustness [TOM: actually, I think for the equivalence with the objective, the $w_{ij}^{(\mathrm{in})}$ should not be the original input weights but those from the previous step. Maybe this is what is meant here anyway?]:

$$w_{ij} \leftarrow w_{ij}^{(\mathrm{in})}\, \frac{Q^2}{w_{ij}^{(\mathrm{in})}\, \Delta_{ij}^2 + Q^2} \tag{16}$$

$$\Delta_{ij} = y_{ij} - \sum_{k=1}^{K} a_{ik}\, g_{kj} \ , \tag{17}$$

where the $w_{ij}^{(\mathrm{in})}$ are the original input (investigator-specified) data weights, Q is the dimensionless soft outlier threshold, and Δ_{ij} is the residual of the (current best-fit) model at datum y_{ij} This formula (16) looks magical but it's just a zero-safe version of a standard iteratively reweighted least squares method [**?**]. As the residual Δ_{ij} gets very large, w_{ij} approaches Q^2/Δ_{ij}^2; as the residual gets small, $w_{ij} \approx w_{ij}^{(\mathrm{in})}$. Note that this weight adjustment only makes sense if the weights have units that are the inverse square of the data units.

The a-step, g-step, SVD reorientation, and w-step are iterated to convergence. Convergence is judged by a dimensionless estimate of the size of the g-step adjustment. The output of training is the full, converged, $N \times K$ matrix A of coefficients a_{ik} and the full, converged, $K \times M$ matrix G of components g_{kj}.

3.1 Objective

In the previous section, the method was described as an algorithm. It turns out that the method can also be described as a maximum likelihood estimator with a heavy-tailed noise model. Without the w-step, the method is just HMF [2], which is a maximum likelihood estimator under a Gaussian likelihood. The objective function is the chi-squared statistic, proportional to the negative log-likelihood:

$$r_{ij} = \frac{y_{ij} - a_i^\top g_j}{\sigma_{ij}}, \qquad \chi^2(A, G) = \sum_{ij} r_{ij}^2 \tag{18}$$

This objective is bilinear in (A, G) so solvable with alternating least squares, as described above. However, the quadratic penalty makes this setup very fragile when there are outliers or anomalies in the data.

To combat this fragility, we added robustness via the w-step. In the context of maximum likelihood estimation, we can instead (equivalently) think of replacing the Gaussian likelihood with a heavy-tailed likelihood, resulting in a sub-quadratic penalty for large residuals, down-weighting their influence. Choosing a Cauchy likelihood, the negative log-likelihood is proportional to our new objective function

$$L(A, G) = \sum_{ij} \frac{Q^2}{2} \log\left(1 + \left(\frac{r_{ij}}{Q}\right)^2\right), \tag{19}$$

where Q is the same dimensionless soft outlier threshold as above. For small residuals $r_{ij} \ll Q$, the penalty is still roughly quadratic, while for large residuals $r_{ij} \gg Q$, the penalty is roughly logarithmic. In Appendix A, we show that including the w-step described above is exactly equivalent to minimizing this objective function.

This objective function is no longer bilinear in (A, G), but it is still biconvex, in that it is convex in A for fixed G and convex in G for fixed A. The existence of the objective means however that we are not actually forced to do the w-step and optimise the latent weights w_{ij}. We could instead directly optimise the objective with modern gradient-based optimizers like Adam or L-BFGS. This would be particularly attractive for very large data where the matrices get huge and the ALS becomes expensive, slow, and memory intensive. In this case, we could use mini-batches stochastic gradient descent to simultaneously optimize A and G. It would be pertinent to include a good reorientation step every $\sim$ 10-100 iterations in this case, due to the degeneracies mentioned above. [TOM: I actually don't know (yet) of a good reorientation steps here that works well for mini-batches and avoids instantiating big matrices. I'm sure something exists but I haven't looked yet.]

Test time: At test time, a new data object y_* with M pixel values y_{*j} is introduced, with associated weights w_{*j}, including probably some missing data with vanishing weights. The a-step and w-step are iterated on this object to convergence, keeping all the components g_{kj} fixed. Convergence is judged by a dimensionless estimate of the size of the a-step adjustment. The output of test time is K converged coefficients a_{*k}, or equivalently the low-rank representation $\sum_k a_{*k}\, g_{kj}$.

Implementation notes Trimming down under-observed objects and wavelengths? Working on residuals not data? Actually output synthetic data not a values?

HOGG: The fastness at training time depends on jax [**?**]. The fastness at test time depends on parallelization. Currently the parallelization isn't correct; need to batch it maybe to reduce shared-memory overheads?

This method does not [HOGG THINKS] optimize a scalar objective; it is *an algorithm*. Therefore, it can't just be put into standard ML form and optimized by contemporary optimization methods like CJ, BFGS, or Adam. [HOGG thinks??] [TOM: I think we can remove this now.]

A Proof of objective

A.1 Auxiliary Form

Claim: the loss can be expressed for any r in the following way

$$\rho(r) = \min_{0<w\leq 1} \left[\tfrac{1}{2} w r^2 + \phi(w)\right], \tag{20}$$

where

$$\phi(w) = \frac{Q^2}{2}\,(w - 1 - \log w). \tag{21}$$

Let's prove that. Define

$$J\,(w; r) = \tfrac{1}{2} w r^2 + \tfrac{Q^2}{2}\,(w - 1 - \log w) \tag{22}$$

Differentiating with respect to w:

$$\frac{\partial J}{\partial w} = \tfrac{1}{2} r^2 + \frac{Q^2}{2}\left(1 - \frac{1}{w}\right), \tag{23}$$

and

$$\frac{\partial^2 J}{\partial w^2} = \frac{Q^2}{2}\frac{1}{w^2} > 0, \qquad \forall\, Q, w > 0. \tag{24}$$

thus we know that the critical point minimises J. Setting $\partial_w J = 0$ yields

$$\hat{w}(r) = \mathrm{argmin}_w\, J(w; r) \tag{25}$$

$$= \frac{1}{1 + (r/Q)^2}, \tag{26}$$

and note that $\hat{w} \in (0, 1]$ because $(r/Q)^2 \geq 0$. We now prove the claim. First set $t = (r/Q)^2 \geq 0$, then

$$\hat{w} = \frac{1}{1+t}, \qquad r^2 = Q^2 t, \tag{27}$$

such that

$$J(\hat{w}; r) = \tfrac{1}{2}\hat{w} r^2 + \phi(\hat{w}), \tag{28}$$

substituting and simplifying one piece at a time:

$$\Rightarrow \tfrac{1}{2}\hat{w}r^2 = \frac{Q^2}{2}\frac{t}{1+t} \tag{29}$$

$$\Rightarrow \phi(\hat{w}) = \frac{Q^2}{2}\left(\hat{w} - 1 - \log\hat{w}\right) \tag{30}$$

$$= \frac{Q^2}{2}\left(-\frac{t}{1+t} + \log(1+t)\right) \tag{31}$$

$$\Rightarrow J(\hat{w}; r) = \frac{Q^2}{2}\log\left(1 + \left(\tfrac{r}{Q}\right)^2\right). \tag{32}$$

Thus

$$\rho(r) = J(\hat{w}; r) \tag{33}$$

$$= \min_{0<w\leq 1}\left[\tfrac{1}{2}wr^2 + \phi(w)\right], \tag{34}$$

as claimed.

A.2 Three-Step Algorithm

Define new objective:

$$J(A, G, W) = \frac{1}{2}\sum_{ij}\left[w_{ij}r_{ij}^2 + \phi(w_{ij})\right], \tag{35}$$

with $r_{ij} = (Y_{ij} - a_i^\top g_j)/\sigma_{ij}$. By construction,

$$L(A, G) = \min_W J(A, G, W), \tag{36}$$

and if

$$\hat{W} = \mathrm{argmin}_w\, J(A, G, W), \tag{37}$$

then

$$\left[\hat{W}\right]_{ij} = \hat{w}(r_{ij}) \tag{38}$$

$$= \frac{1}{1 + (r_{ij}/Q)^2}. \tag{39}$$

This immediately yield's Hogg's procedure

w-step:	$w_{ij} \leftarrow \hat{w}(r_{ij})$,
a-step:	solve WLS for A with new weights,
g-step:	solve WLS for G with new weights.

where the a-step optimises the quadratic

$$Q(A \,|\, G, W) = \frac{1}{2}\sum_{ij} w_{ij}r_{ij}^2, \tag{40}$$

and the g-step optimises $Q(G \,|\, A, W)$. It should be pretty apparent now that the procedure gives the MLE with a Cauchy likelihood.

A.3 Extra convincing (showing that the procedure optimises L)

Consider one outer cycle starting at $(A^{(t)}, G^{(t)})$. Choose $W^{(t)} = \hat{w}(r(A^{(t)}, G^{(t)}))$. Then

$$L(A^{(t)}, G^{(t)}) = J(A^{(t)}, G^{(t)}, W^{(t)}). \tag{41}$$

With frozen $W^{(t)}$, the a- and g-steps minimize $Q(\cdot \,|\, W^{(t)})$. Since our total objective is $J = Q + \sum \phi(W^{(t)})$, this implies

$$J(A^{(t+1)}, G^{(t+1)}, W^{(t)}) \leq J(A^{(t)}, G^{(t)}, W^{(t)}). \tag{42}$$

We're guaranteed to be helped by the w-step again now, so setting

$$W^{(t+1)} = \hat{w}\left(r(A^{(t+1)}, G^{(t+1)})\right), \tag{43}$$

and using our result from the previous section gives

$$J(A^{(t+1)}, G^{(t+1)}, W^{(t+1)}) \leq J(A^{(t+1)}, G^{(t+1)}, W^{(t)}). \tag{44}$$

Thus chaining the inequalities and $L(A, G) = \min_W J(A, G, W)$ gives

$$L(A^{(t+1)}, G^{(t+1)}) \leq L(A^{(t)}, G^{(t)}). \tag{45}$$

This is enough to guarantee that robust HMF with Hogg's w-step converges to the Cauchy MLE.

A.4 Generalisation

This procedure can be generalised to other likelihoods, their associated $\rho(r)$ functions, and their associated $\phi(w)$ functions. The only requirement is that the $\rho(r)$ function can be expressed in the auxiliary form above, such that the update rule can be derived. This will be generally true for any $\rho(r)$ that is symmetric, monotonically increasing in $|r|$, and sub-quadratic [TOM THINKS]. It might be worth actually writing down this whole thing in terms of student's t-distribution, which is a generalisation of both the Cauchy and Gaussian distributions. Student's t-distribution also comes with the nice interpretation of marginalising out an unknown variance with a inverse Gamma prior, where actually the update rule is the posterior mean of the variance given the data and the prior, such that the w_{ij} weights are latent inverse variances in a now hierarchical model with a Gaussian likelihood [**?**]. Of course, the choices here already have this interpretation, because the Cauchy is a student's t with one degree of freedom, and so the latent variance has an inverse Gamma prior with one degree of freedom. [TOM: as is this is under-explained and underdeveloped, but I think it's a good idea. I find the idea that the weights are latent variances in a hierarchical model appealing, since we can still interpret the method as having a Gaussian likelihood, just with unknown variances that we marginalise out (with a inverse Gamma prior on the variances or Gamma on the precisions). This *feels* more natural that just saying we have a Cauchy likelihood (I don't expect the data to actually be Cauchy distributed), but this is just interpretation in the

end anyway.]

References

[1] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

[2] P Tsalmantza and David W Hogg. A data-driven model for spectra: Finding double redshifts in the Sloan Digital Sky Survey. *The Astrophysical Journal*, 753(2):122, 2012.