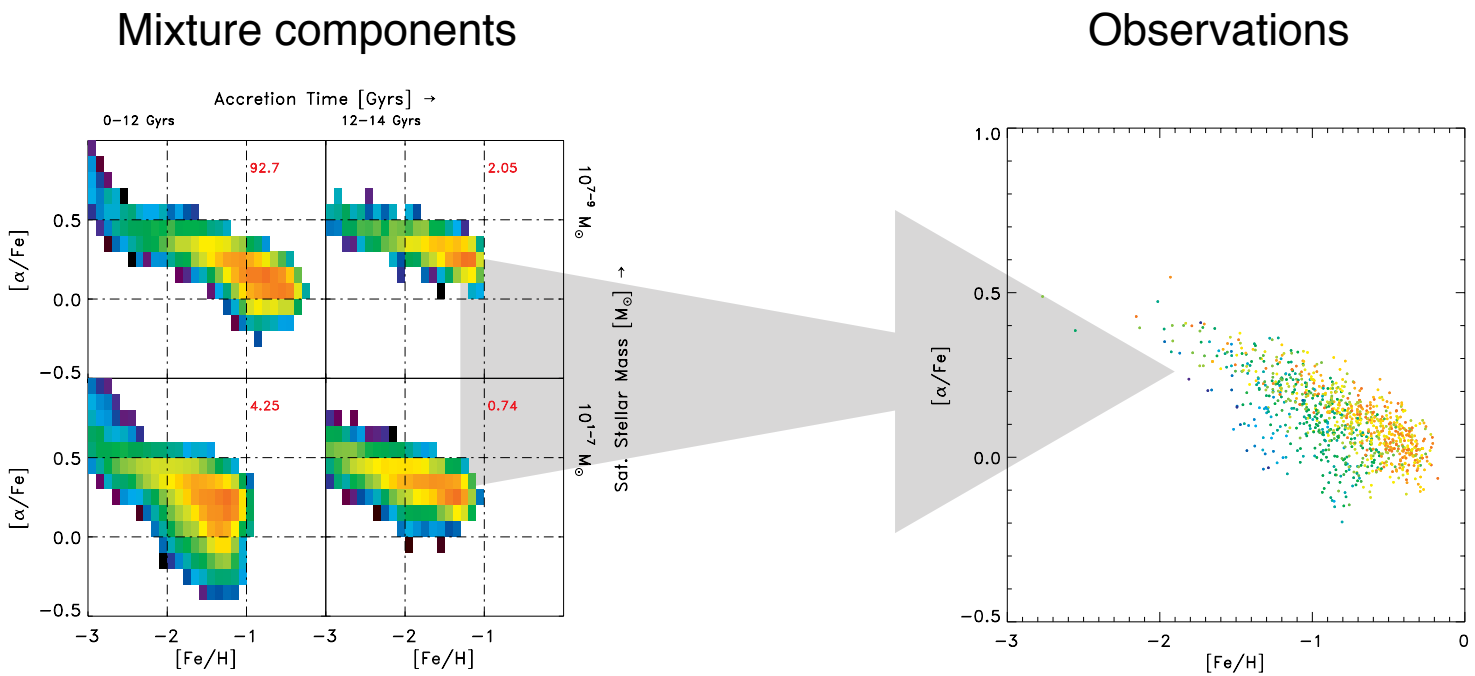
A generative finite mixture model



$$\left[\frac{Fe}{H}, \frac{\alpha}{Fe}\right]_{i=1}^{N} \text{ i.i.d} \sim f(x,y) = \sum_{j=1}^{m} \pi_j f_j(x,y)$$

Where the mixing proportions, π, give the formation history.

A generative finite mixture model

- in order to recover the formation history of the halo
- we propose a generative model in the form of a finite mixture model
- where each observed point comes from one of m mixture components (pictured)
- since each mixture component has an associated mass and accretion time range, the formation history is specified if we know what percentage of observations come from each mixture component
- our goal, then, is to determine the mixing proportions, π
- note that the observed points we are attempting to fit are generated via simulation, not our proposed generative model
- we don't know the true generative method, and are proposing a reasonable one that we think can be fit, and that will reveal the formation history

Model definition

$$\text{Let} \qquad x = \frac{\alpha}{Fe}, \qquad y = \frac{Fe}{H}$$

Given m mixture components, we propose that the density from which observations are generated is

$$f(x, y) = \sum_{j=1}^{m} \pi_j \, f_j(x, y) \tag{1}$$

- Mixing proportion
- Mixture component j

$$\text{where} \qquad \sum_{j=1}^{m} \pi_j = 1, \qquad \pi_j \geq 0, \qquad j = 1, \ldots, m$$

Definitions

- For notational simplicity, x=$\frac{\alpha}{Fe}$ and y=$\frac{Fe}{H}$
- Formally, given m mixture components, the density from which all observations are drawn is as shown
- each observation comes from one of the mixture components with some probability π_j
- The mixing proportions, $\boldsymbol{\pi}$ must be non-negative, and sum to 1
- the mixture components, f_j, are known and taken as given

Estimating the mixing proportions π

To estimate the mixing proportions, we can use a maximum likelihood approach

$$\hat{\pi}_{\text{MLE}} = \underset{\pi}{\text{argmax}}\, \mathcal{L}(\pi)$$

$$\text{where} \qquad \mathcal{L}(\pi) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{m} \pi_j f_j(x_i, y_i) \right)$$

Unfortunately the standard MLE procedure for estimating π is intractable with this likelihood.

The Expectation Maximization (EM) algorithm provides an alternative way to estimate $\hat{\pi}_{\text{MLE}}$

Estimating the mixing proportions π

- 1

Expectation Maximization

Suppose we knew which mixture component f_j each observation came from:

$$z_{ij} = \mathbf{1}(x_i, y_i \sim f_j) = \begin{cases} 1 & (x_i, y_i) \sim f_j \\ 0 & \text{otherwise} \end{cases}$$

The log likelihood can then be expressed as

$$\ell(\boldsymbol{\pi}) = \sum_{i=1}^{n} \sum_{j=1}^{m} z_{ij} \log \left\{ \pi_j f_j(x_i, y_i) \right\}$$

The addition of the latent variable $\mathbf{z}$ actually makes things easier because it is easily differentiable in $\boldsymbol{\pi}$.

Expectation Maximization

- Suppose we knew which mixture component f_j each observation came from
- Then we could construct a latent indicator variable, z_{ij}, which is 1 if point i comes from mixture component j, and 0 otherwise
- The log like then becomes
- Since we're supposing that we know z_{ij}, it's trivial to differentiate this log likelihood with respect to $\hat{\pi}$
-

Estimating $\hat{\pi}$ using expectation maximization

We don't know $\mathbf{z}$, so we replace $\mathbf{z}$ with the expected value of $\mathbf{z}$, conditioned on the data and the last known $\hat{\pi}$:

$$\hat{\pi}^{(t)} = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}\left[\ell(\pi)\middle|\mathbf{x}, \mathbf{y}, \hat{\pi}^{(t-1)}\right]$$

Starting with some random initial value for $\hat{\pi}^{(0)}$, we iteratively

- Find the expected value of $\ell(\pi)$ using the current expected values of the latent variable $\mathbf{z}$
- Set $\hat{\pi}^{(t)}$ to the $\underset{\pi}{\operatorname{argmax}}$ of this expectation, which is simple to compute

And repeat until $\ell(\pi)$ stabilizes to a range $< 10^{-4}$

Estimating the mixing proportions π

- We don't know $\mathbf{z}$, so we replace $\mathbf{z}$ with the expected value of $\mathbf{z}$, conditioned on the data and the last known $\hat{\pi}$:
-
- the true likelihood is increasing in each iteration

Find the expected value of $\ell(\boldsymbol{\pi})$ using the current expected value of the latent variable

The expected value of $\ell(\boldsymbol{\pi})$, with respect to the conditional distribution of $\mathbf{z}$, given observed data and $\hat{\boldsymbol{\pi}}^{(t-1)}$ is

$$\mathbb{E}_{\boldsymbol{\pi}}\left[\ell(\boldsymbol{\pi})\middle|\mathbf{x},\mathbf{y},\hat{\boldsymbol{\pi}}^{(t-1)}\right] = \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}_{\boldsymbol{\pi}}\left[z_{ij}|x_i,y_i\right]\left\{\log f_j(x_i,y_i) + \log \pi_j\right\}$$

Since z_{ij} is an indicator, its expected value is simply the probability that data point i comes from model j

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\pi}}\left[z_{ij}|x_i,y_i\right] &= \Pr_{\boldsymbol{\pi}}(z_{ij}|x_i,y_i)\\ &= \frac{p(x_i,y_i|z_{ij}=1)p(z_{ij}=1)}{p(x_i,y_i)}\\ &= \frac{\pi_j f_j(x_i,y_i)}{\sum_{j=1}^{m}\pi_j f_j(x_i,y_i)}\end{aligned}$$

Find the expected value of $L(\pi)$ using the current expected value of the latent variable

- cond prob

Find the $\underset{\boldsymbol{\pi}}{\text{argmax}}$ of this expectation

Now that we have the expected value of $\ell(\boldsymbol{\pi})$ with respect to the conditional distribution of $\mathbf{z}$, we need only evaluate

$$\hat{\boldsymbol{\pi}}^{(t)} = \underset{\boldsymbol{\pi}}{\text{argmax}}\, \mathbb{E}\left[\ell(\boldsymbol{\pi}) \middle| \mathbf{x}, \mathbf{y}, \hat{\boldsymbol{\pi}}^{(t-1)}\right]$$

Which can be analytically specified, at each time t, as:

$$\hat{\pi}_k^{(t)} = \frac{\sum_{i=1}^{n} w_{ij}^{(t-1)}}{n}$$

where $$w_{ij}^{(t+1)} = \frac{\pi_j^{(t)} f_j(x_i, y_i)}{\sum_{k=1}^{m} \pi_k^{(t)} f_k(x_i, y_i)}$$

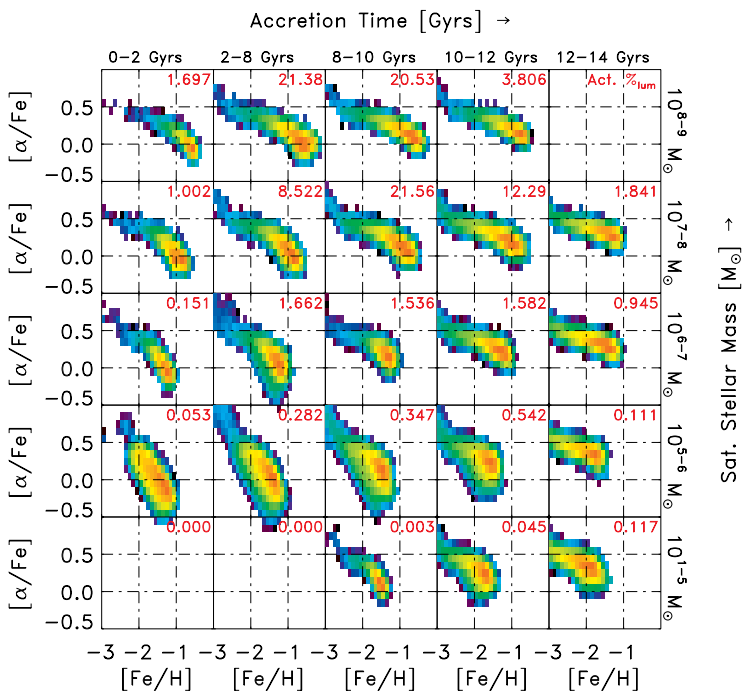
Find the argmax of this expectation

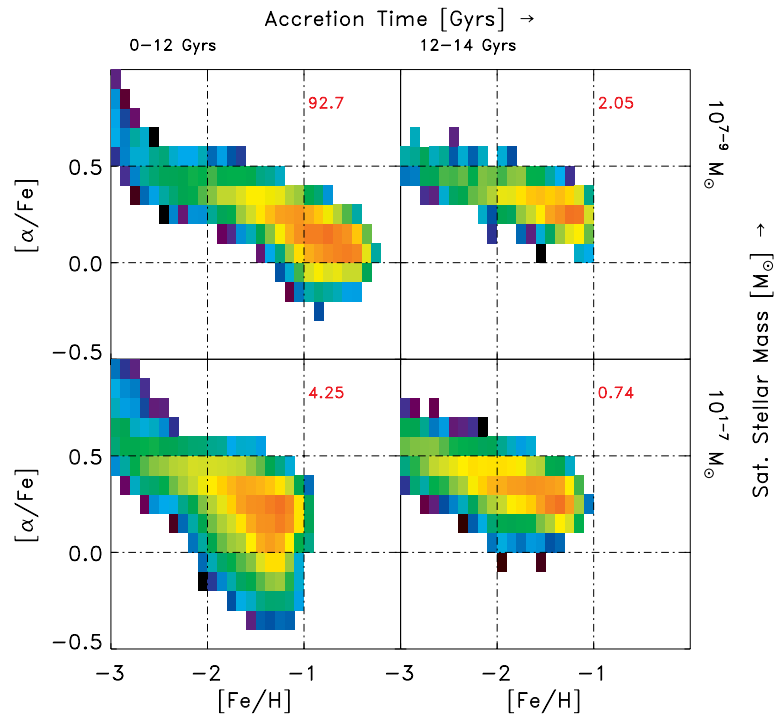
$$\pi$$

- Note how simple this is to compute

Simulations

- Generated observations from 10 realizations of halos
- Generated mixing components for these halos
- Used a 5x5 grid ($m = 25$), and several 2x2 grids ($m = 4$)



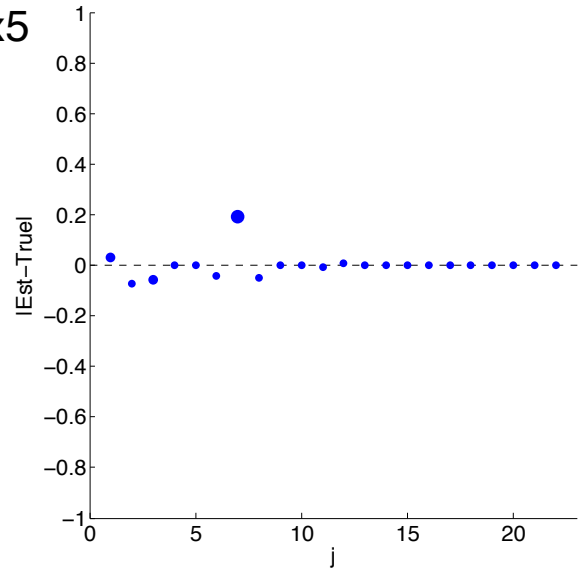


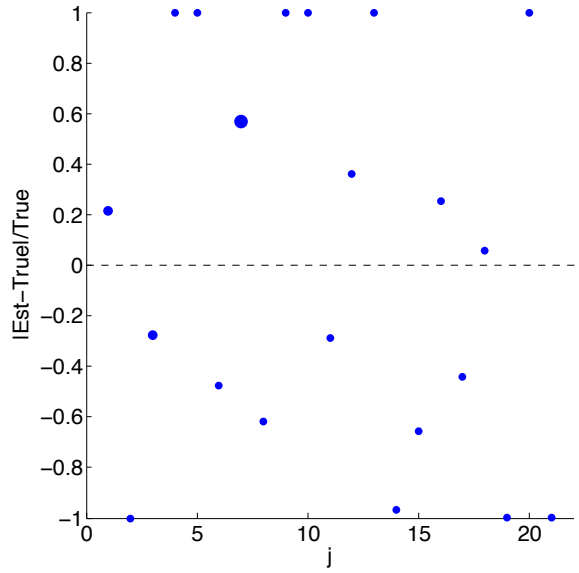
Simulations

- Tried several different mass/accretion time separations for 2x2
- 5x5 grid did not work for some halo realizations
- 2x2 grid reliably converged on the correct mixing proportions
- since these observations were generated from the simulations (not our model) we know the correct mixing proportions, pi

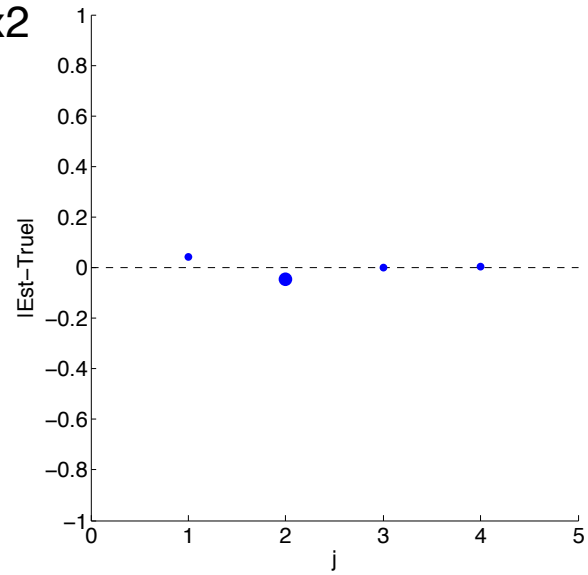
Halo 2: least accurate 2x2 fit

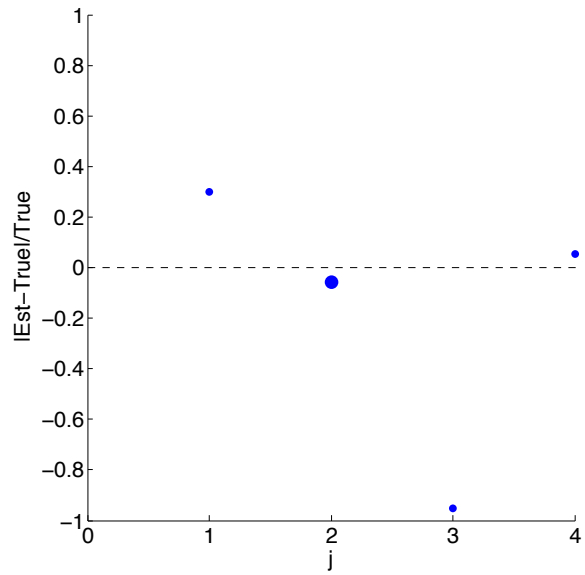
5x5





2x2



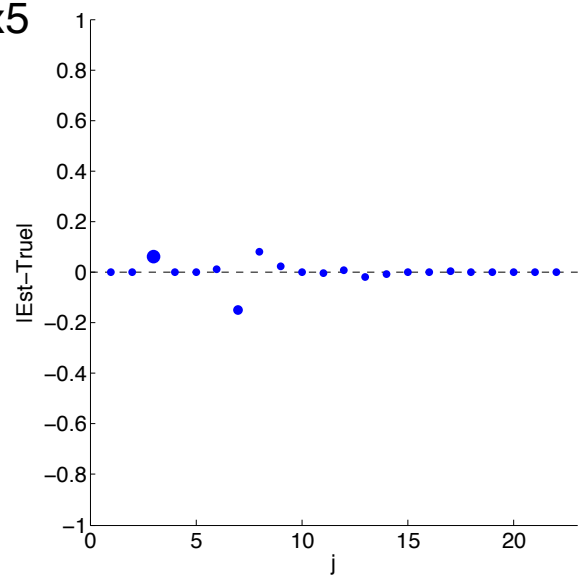


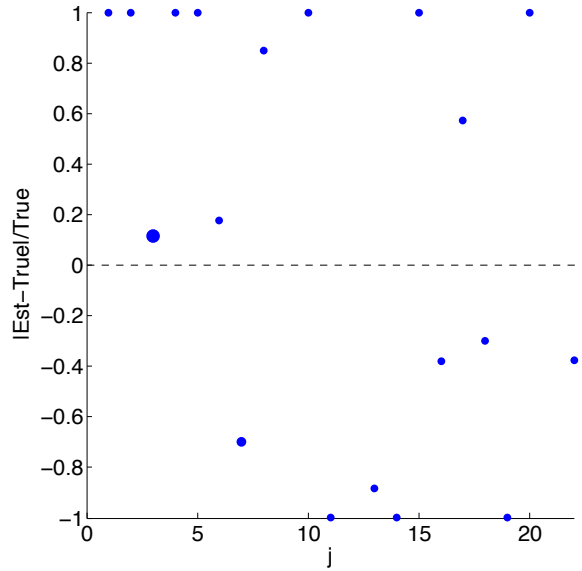
Halo 2 results

- top is 5x5
- bottom is 2x2
- left is absolute difference between the true mixing proportions and the estimated ones
- right is percent difference between the true mixing proportions and the estimated ones
- 2x2 is pretty close, although not a perfect fit, it does reconstruct the formation history fairly accurately.

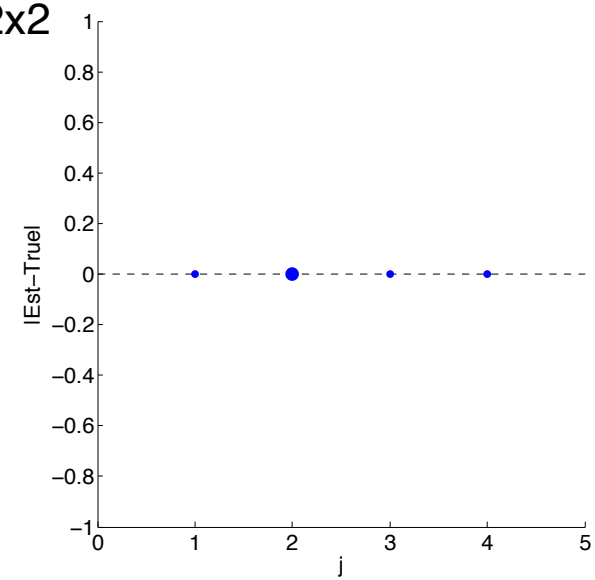
Halo 5: most accurate 2x2 fit

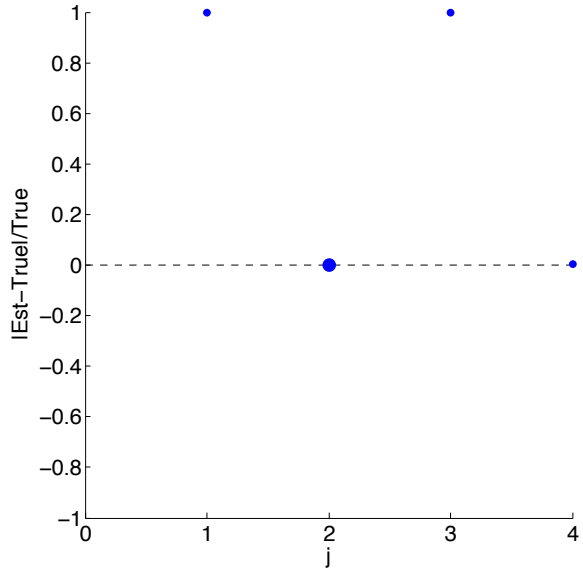
5x5





2x2

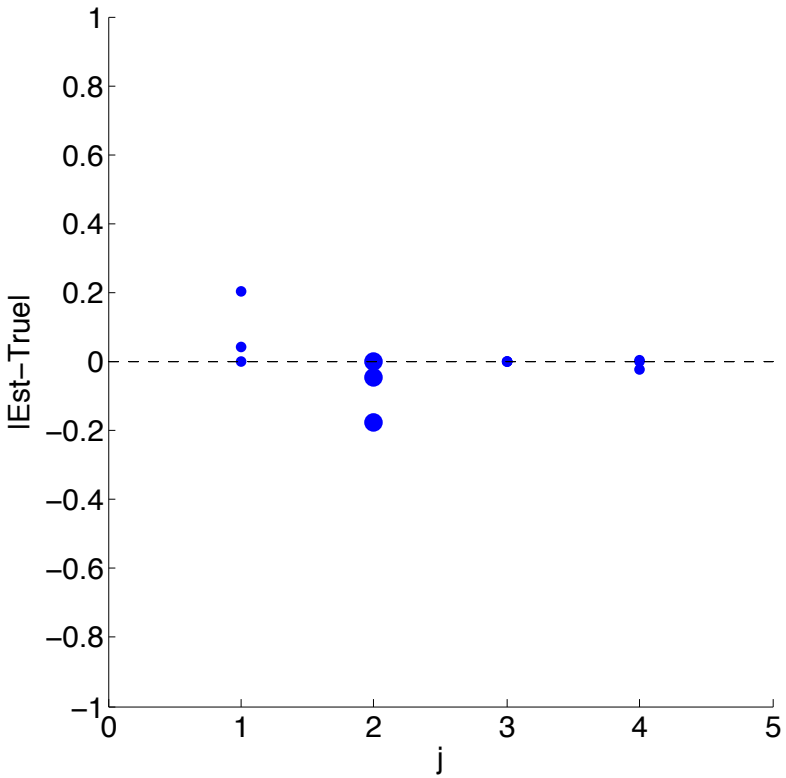


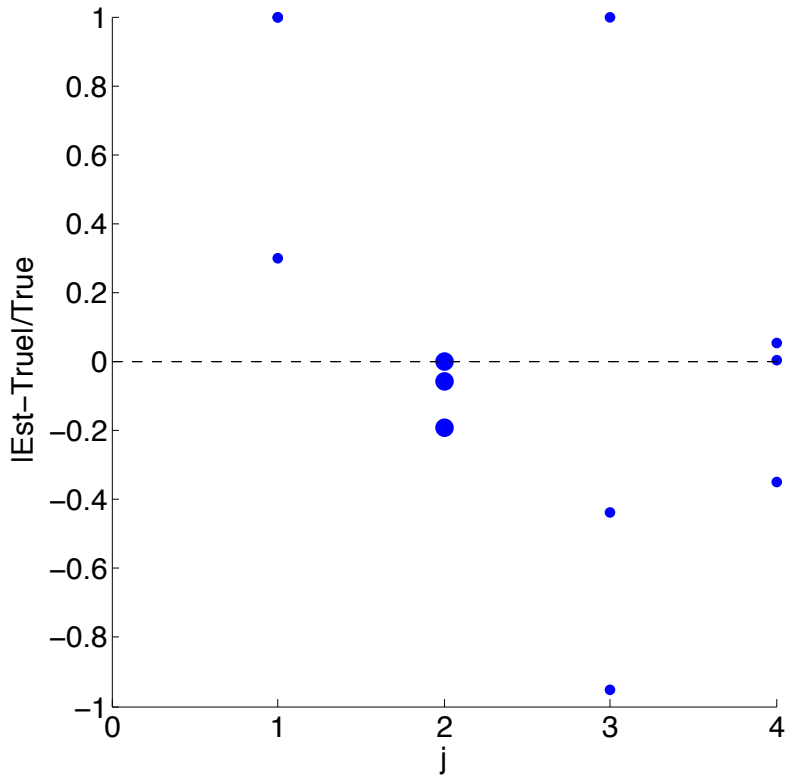


Halo 5 results

- Running the same algorithm for another halo realization
- In this case, the 2x2 is dead-on
- The large percent differences are caused by a true value of 0 and an estimated value of $< 1x10^{-6}$
- In general, the 10 different halo realizations have accuracies falling somewhere in between the good fit pictured here, and the less good fit from the previous slide.

2x2: All 10 halos



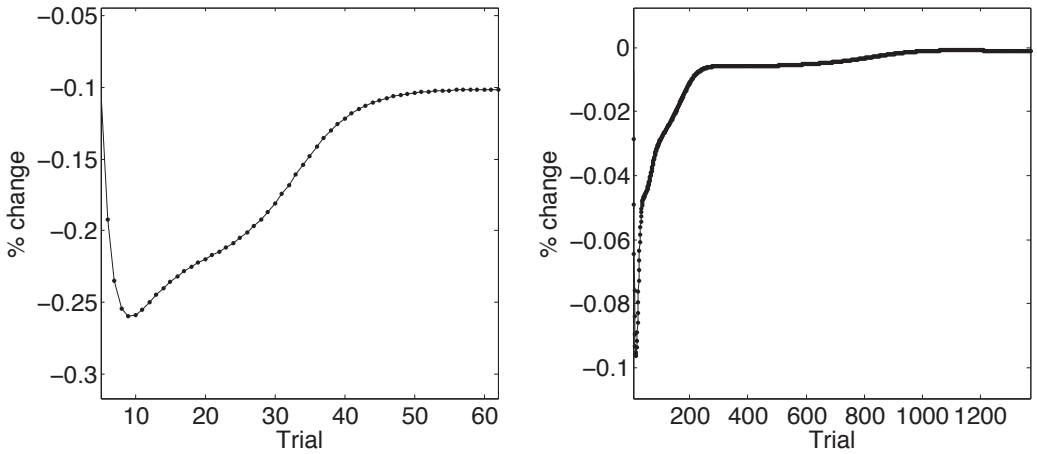


2x2: All 10 halos

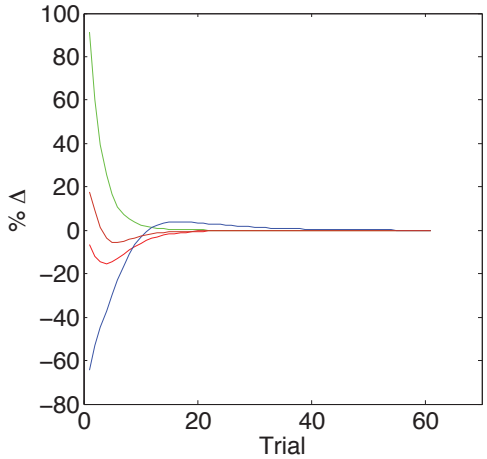
- Same graphs, but all 10 halo realizations stacked on top of eachother
- In general, the 2x2 reconstructs the formation history fairly well

Convergence and minimum observation size

Log likelihood % change



π % change



- Works with as few as 1,000 observations
- Insensitive to initialization of π
- Large weights identified after 10 iterations
- $\ell(\pi)$ stops changing appreciably after 60 (m=4) or 600 (m=25) iterations
- Always converges

Convergence and minimum observation size

- The EM algorithm is not known to converge particularly quickly
- For a 2x2 with 1,000 observations, it takes about 60 iterations, or 0.3 seconds, to trigger the stopping condition of less than 10^{-4} change in log likelihood
- since our mixture model is not the true generative model, we can consistently converge on estimates of the mixing proportions that are incorrect.
- this happened with 5x5 grids especially, and could be a reflection of over-fitting, or a symptom of too-tight a grid
- we did not see any degeneracies–we always converged on the same answers, even if they weren't the "true" values.

Covariance and confidence intervals

The asymptotic covariance matrix of $\hat{\boldsymbol{\pi}}$ can be approximated by the inverse of the observed Fisher information matrix, I:

$$I(\boldsymbol{\pi}'|\mathbf{x},\mathbf{y}) = -\frac{\partial^2 \ell(\boldsymbol{\pi}')}{\partial \boldsymbol{\pi}' \partial \boldsymbol{\pi}'^T}$$

$$\mathrm{Cov}(\hat{\pi}_p, \hat{\pi}_q) = \left[I^{-1}(\hat{\boldsymbol{\pi}}')\right]_{pq}$$

with variance and correlation given by

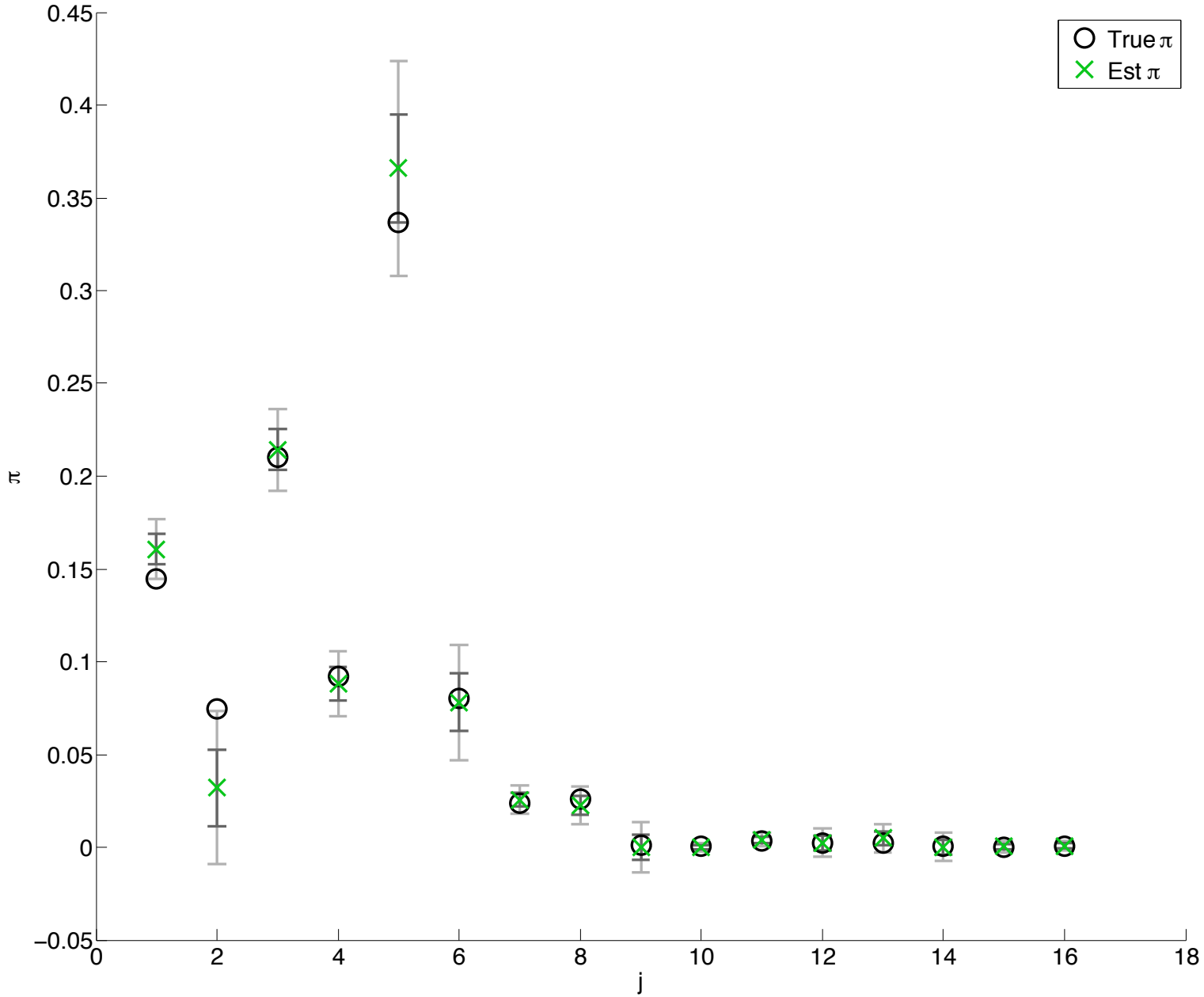
$$\mathrm{Var}(\hat{\pi}_j) = \sigma_j^2 = \left\{\mathrm{Cov}(\hat{\boldsymbol{\pi}})\right\}_{jj}$$

$$\mathrm{Corr}(\hat{\pi}_p, \hat{\pi}_q) = \frac{\mathrm{Cov}(\hat{\pi}_p, \hat{\pi}_q)}{\sqrt{\sigma_p^2 \sigma_q^2}}$$

Covariance and confidence intervals

- 1

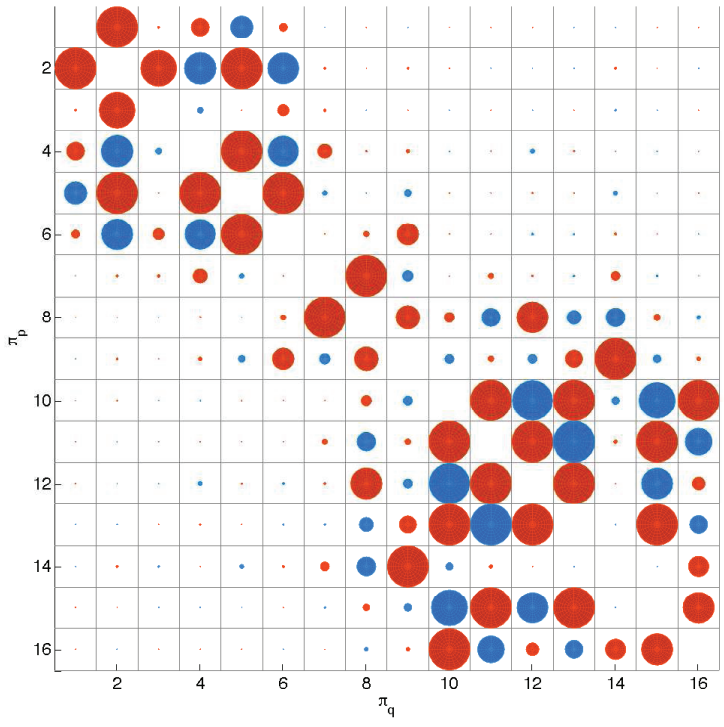
Confidence Intervals

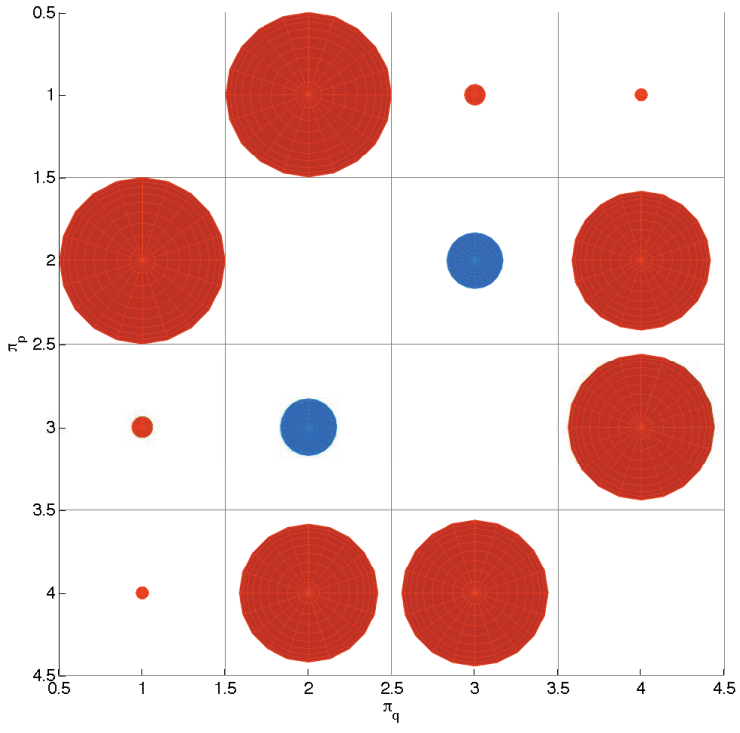


Confidence Intervals

- Using m-out-of-n bootstrapping, at a 95% confidence level, produced similar results
- for 5x5 grids the estimates of the mixing proportions were not always inside 2 standard deviations, or the 95% bootstrapped confidence interval
- for 2x2 grids, the estimates are almost always inside 2 standard deviations, or the 95% bootstrapped confidence interval

Correlation between π





Correlation between π

- larger spheres represent higher values of pi hat
- red spheres are negative correlation
- blue spheres are positive correlation
- mixing components that are right next to eachother, and thus have similar mass and accretion time ranges, tend to have larger correlations.
- one way to reduce the correlation might be to produce mixing components on different grids of mass and accretion time
- Again, m-out-of-n bootstrapping produced similar covariance structures

Conclusion

Poor results

- 5x5 grids, except in a few cases
- Parametric bootstrapping

Good results

- 2x2 grids
- EM
- 5x5 in a few cases
- Confidence intervals
  - Observed Fisher information
  - M-of-n bootstrapping

Future work

- Adaptive partitioning of mass and time since accretion for mixing components
- Smoothing the 1,500 metallicity curves and constructing mixing components from them

Conclusion

- starting with what didn't work
- 5x5 grids tended to converge to the wrong answers in all but a handful of cases
- since our mixture model is not the true generative model, we can consistently converge on estimates of the mixing proportions that are incorrect.
- this is also likely the reason that parametric bootstrapping didn't work–the observed points are not actually generated from a set of mixing components, and so when we make the grid too granular, we find patterns that are not found in the simulated data.

- we did have success with 2x2 grids
- the EM algorithm converged, and we did not see any degeneracies–we always converged on the same answers, even if they weren't the "true" values.
- in some cases, the 5x5 grids were fit quite well
- m-out-of-n bootstrapping confirmed confidence intervals and covariance matrices derived from observed Fisher information

- we have done some work on adaptive gridding, based on finding the partitions of mass and time since accretion that maximize the difference between the mixing components, and the underlying metallicity curves
- adaptive gridding looks promising, but is still in the early stages
- since we have 1,500 metallicity curves, we have also investigated smoothing the curves to create 1,500 mixture components
- this might increase sensitivity of the algorithm
- this approach is also compatible with adaptive gridding