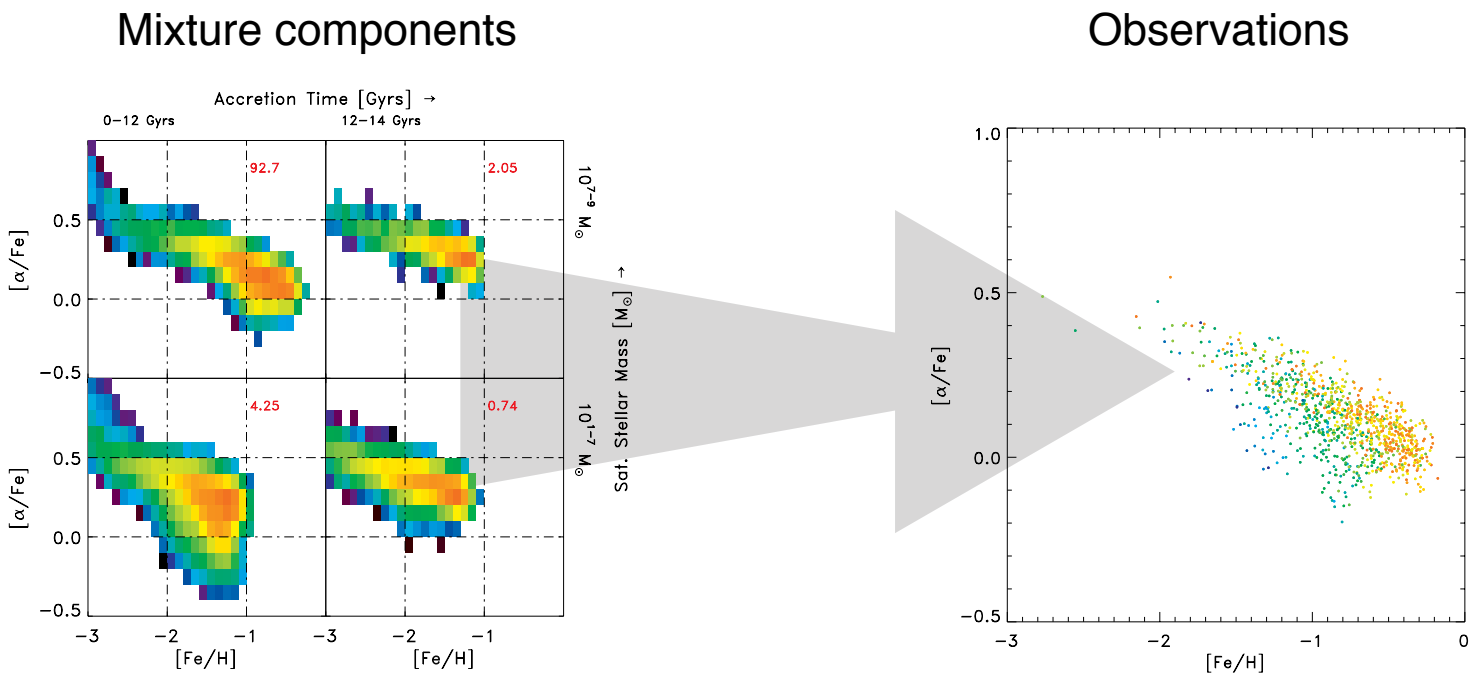
A generative finite mixture model



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

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 (the case of m=4 is pictured on the left)
- 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
- the observed points we are attempting to fit are based on some subset of the 1,500 satellites from simulations, whereas the mixture components are based on all 1,500 satellites
- our generative model is designed to be generic enough to fit specific halos using a more general set of templates.

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

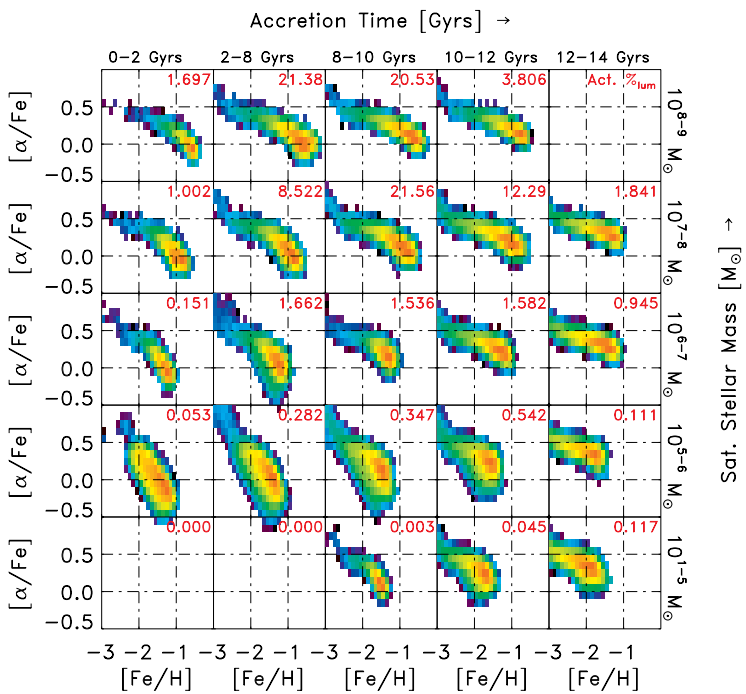
$$\text{where} \qquad \sum_{j=1}^{m} \pi_j = 1, \qquad \pi_j \geq 0, \qquad j = 1, \ldots, m$$

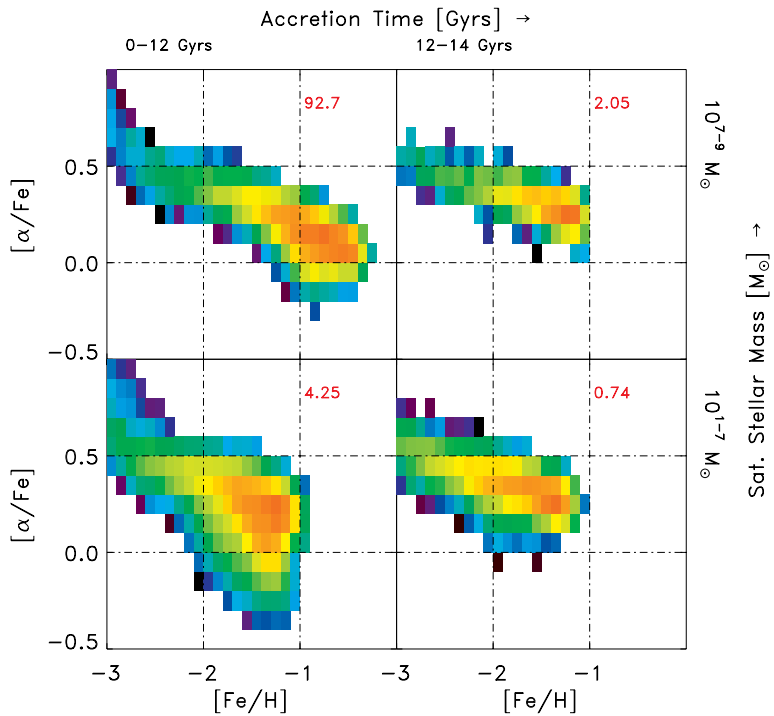
Definitions

- 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

Simulations

- Generated observations from 11 realizations of halos
- Generated mixing components for these halos
- Used a 5x5 grid ($m = 25$), and several 2x2 grids ($m = 4$)

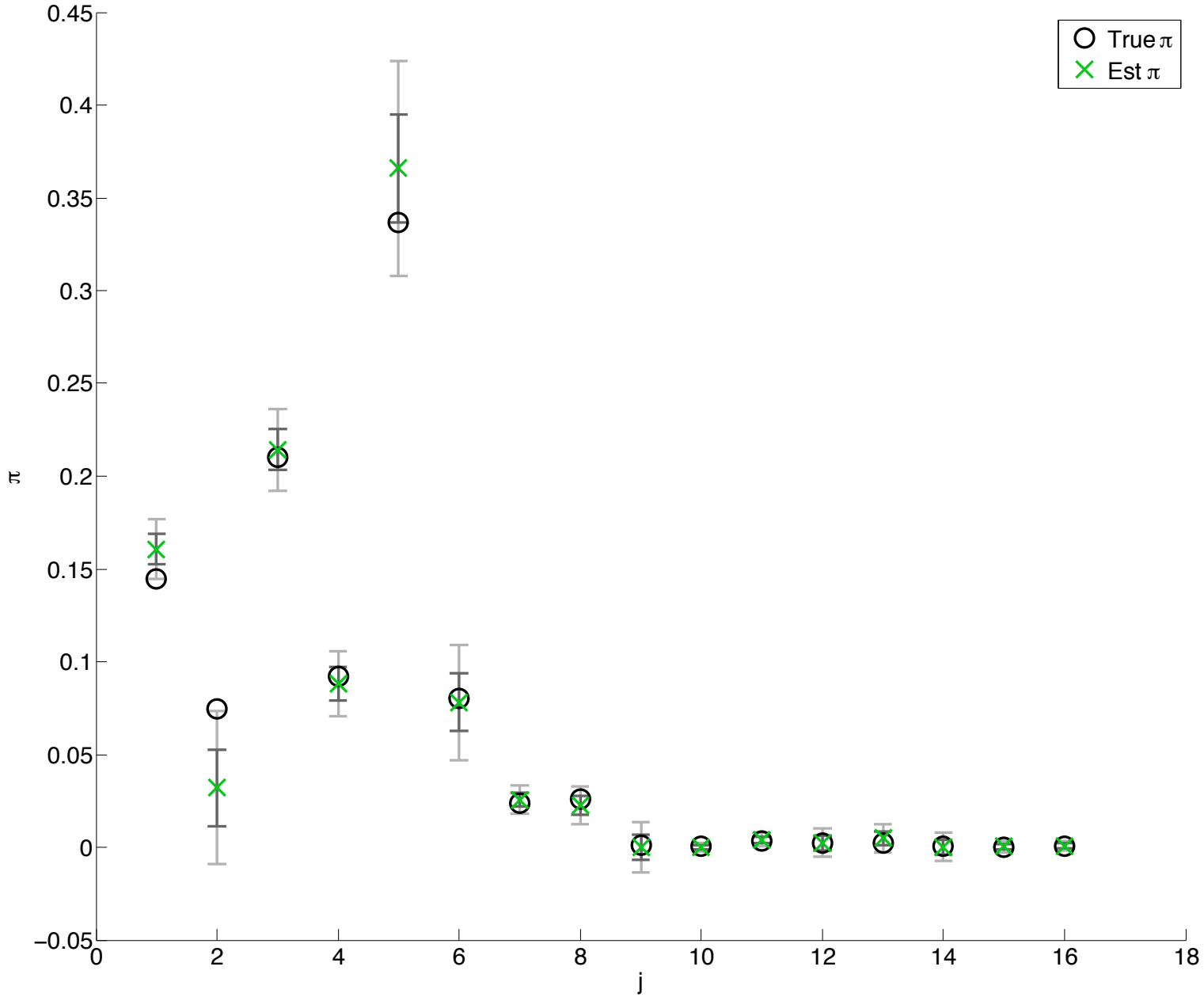




Simulations

- Tried several different mass/accretion time separations for 2x2
- 2x2 grid reliably converged on the correct mixing proportions
- 5x5 grid produced an accurate formation history for some halo realizations, but not all
- since the mixing components are generated from 1,500 satellites, and the observed data points from some subset of these, the 5x5 grid was too granular–there is a limit to how specific we can be if our generative model doesn't know about the vagaries of the individual galaxy its looking at
- since these observations were generated from the simulations (not our model) we know the correct mixing proportions, pi

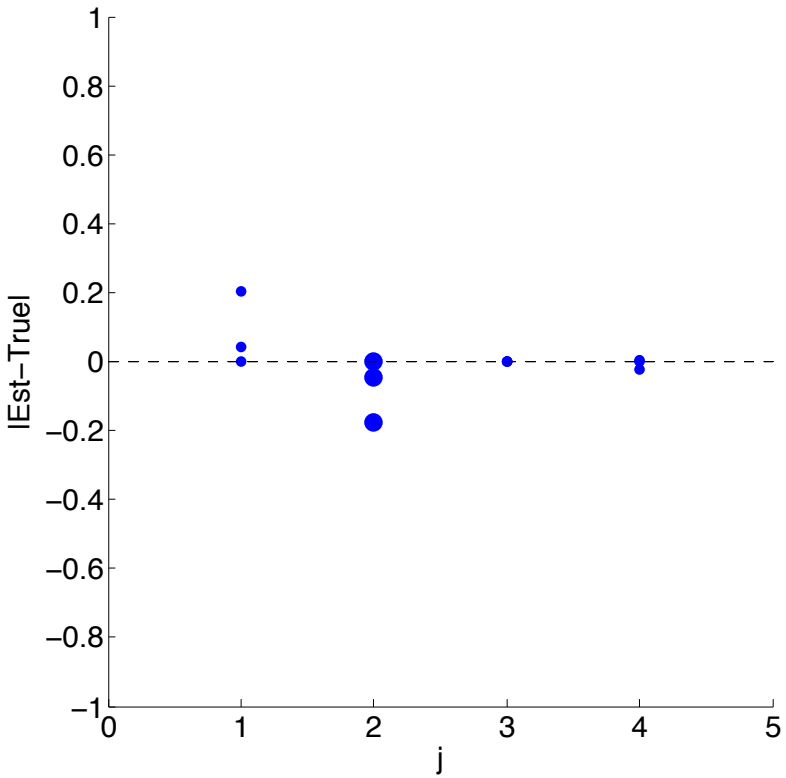
Results for a 5x5 grid ($m = 25$), for one halo realization

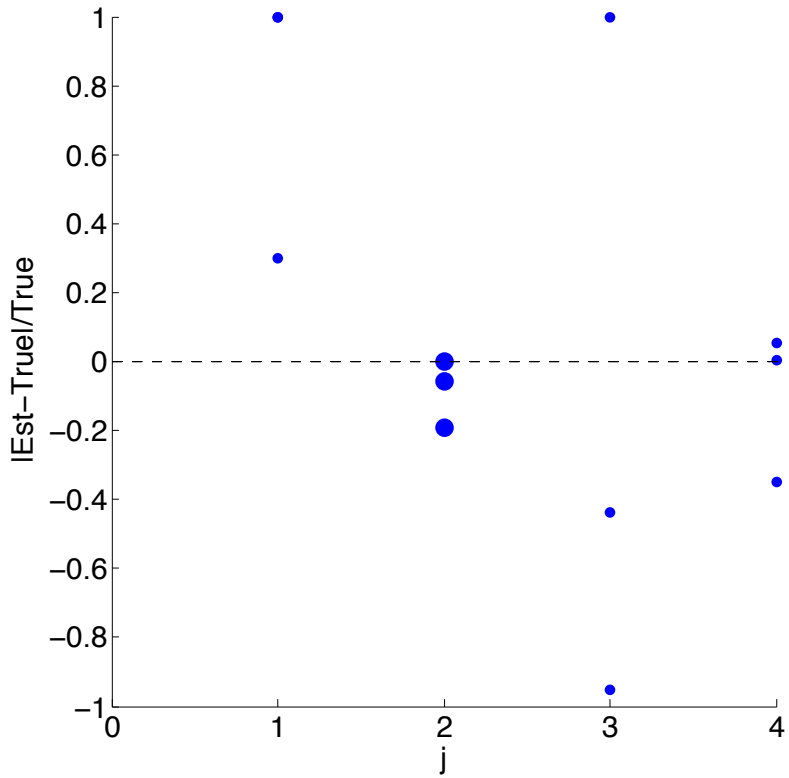


Simulations

- Note the good fit
- Fits even small mixing proportions, e.g. 1% of total luminosity
- We got these results by implementing the aforementioned finite mixture model using the expectation maximization algorithm

2x2: All 11 halos





2x2: All 11 halos

- Absolute difference btwn true and est on left
- Showing all 11 halos, or 44 points
- In general, the 2x2 reconstructs the formation history fairly well
- Most errors are very small, and none are orders of magnitude off

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

- MLE is a common statistical approach
- give we have some densities, say Normal(mu, sigma), and some data, we choose the mu and sigma that maximize the joint probability, or likelihood, of the data we observe.
- Typically the log of the likelihood is used because it's easier to work with.
- In our case, this means that pi hat, or the estimate of the mixing proportions, is the argmax of the log likelihood.

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

- given that we know the mixture components, or fj's
- 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 complete log likelihood can then be expressed as
- 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

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

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:

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

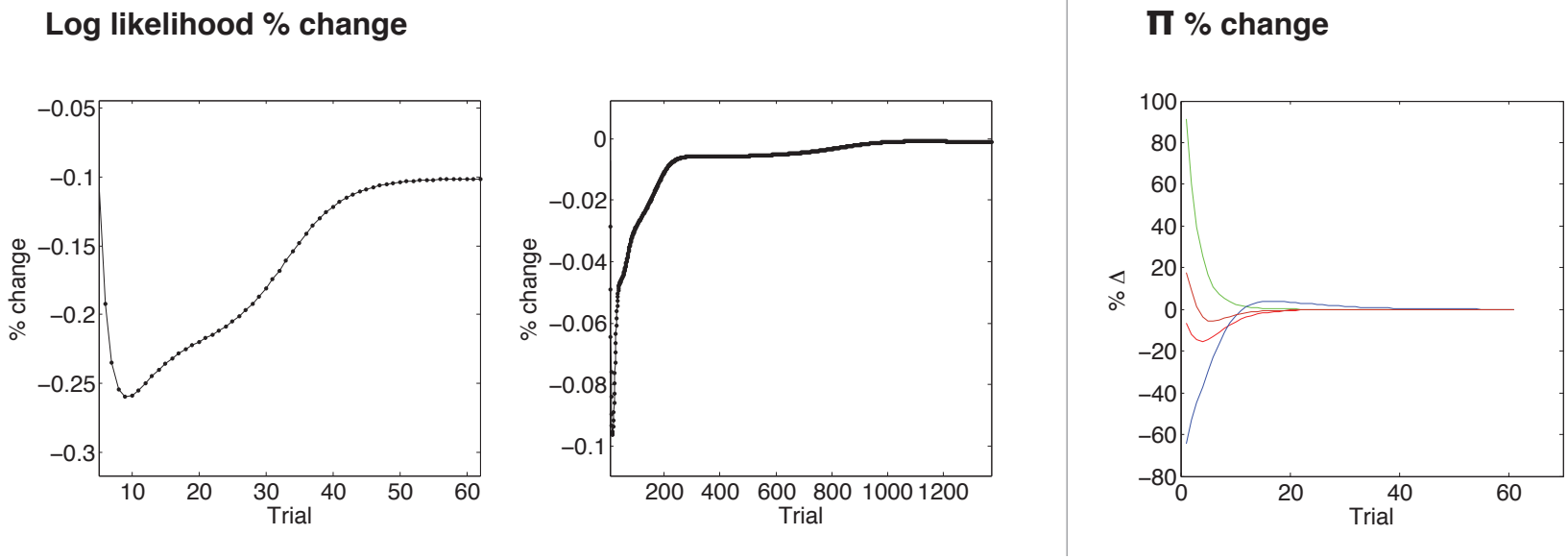
where $$w_{ij}^{(t+1)} = \frac{\pi_j^{(t)} f_j(x_i, y_i)}{\sum_{k=1}^{m} \pi_k^{(t)} f_k(x_i, y_i)}$$

Find the argmax of this expectation

$$\pi$$

- Note how simple this is to compute

Convergence and minimum observation size



- Produces reasonable results few as 1,000 observations
- Confidence intervals narrow with more data
- 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

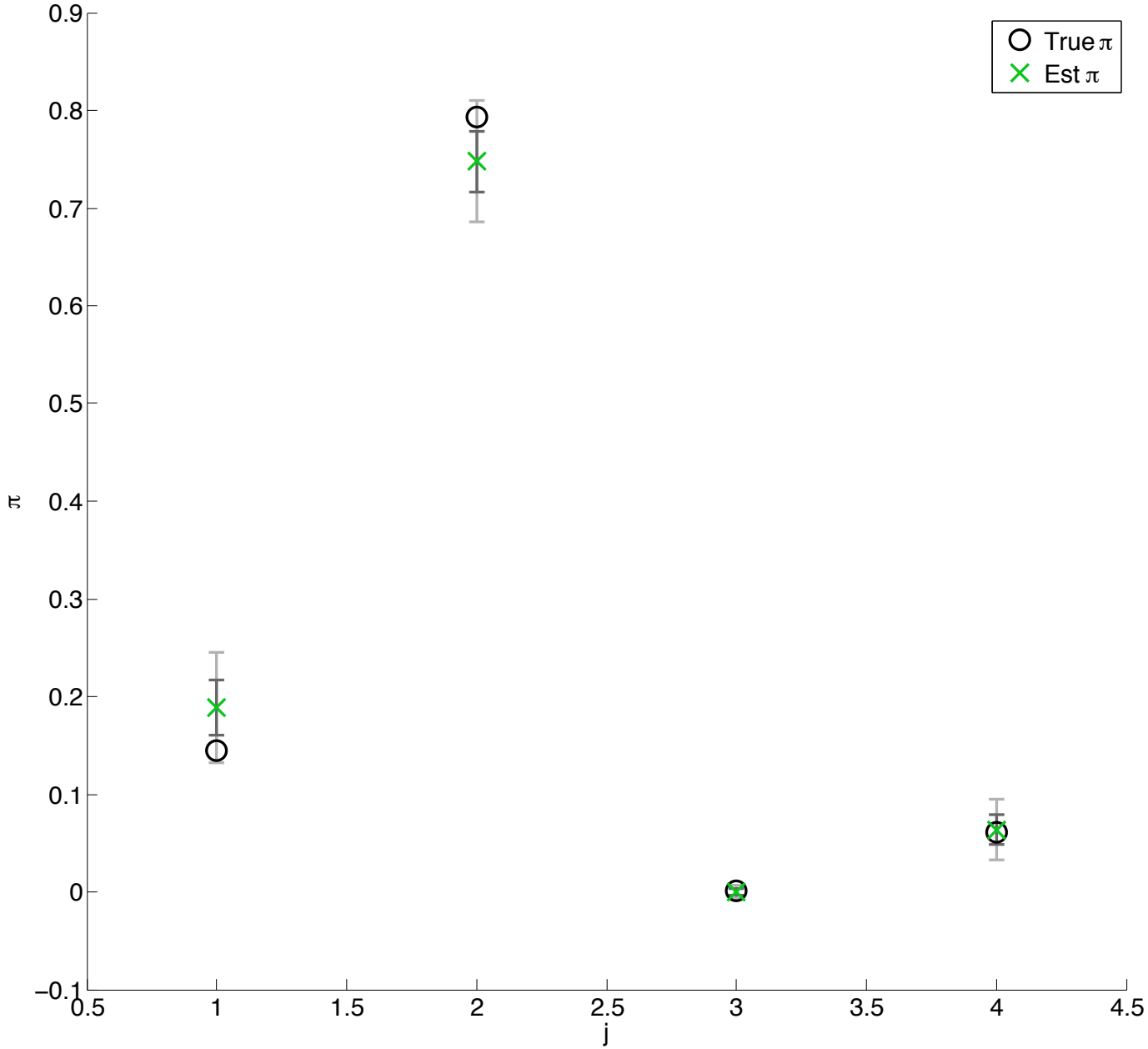
$$\mathrm{Var}(\hat{\pi}_j) = \sigma_j^2 = \left\{\mathrm{Cov}(\hat{\boldsymbol{\pi}})\right\}_{jj}$$

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

Covariance and confidence intervals

- 1

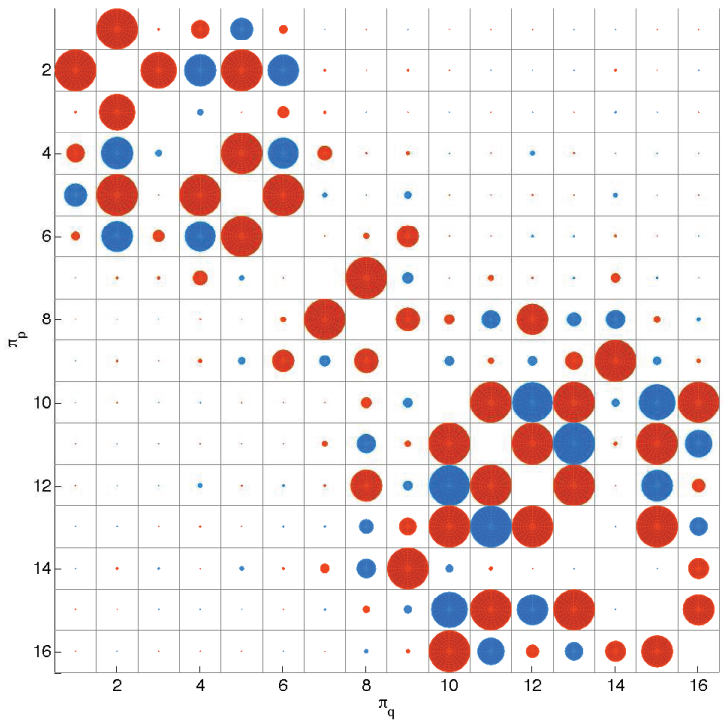
Confidence Intervals: 2x2 results

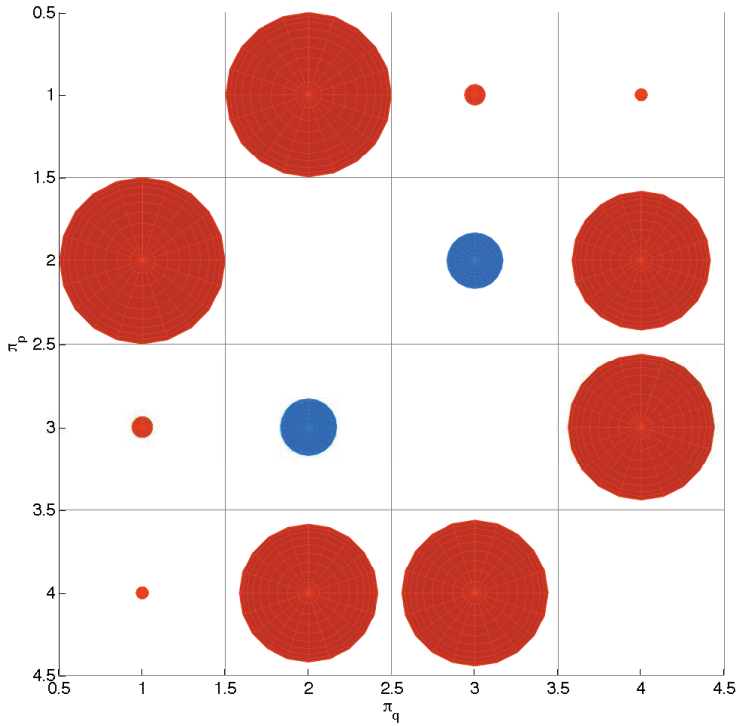


Confidence Intervals

- Using n-out-of-n bootstrapping, at a 95% confidence level, produced similar results
- for 2x2 grids, the estimates are almost always inside 2 standard deviations, or the 95% bootstrapped confidence interval

Correlation between $\hat{\pi}$





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

- We were able to reconstruct the formation history
  - For multiple halo realizations
  - With a single finite mixture model
  - With good accuracy on a 2x2 grid
  - Relatively quickly
  - Equally well for large and small values of π_j
  - With more data, more granular grids could be used
- We found confidence intervals and covariance matrices for the mixing proportions
  - Fisher information and n-out-of-n bootstrapping produced nearly identical results

Future work

- Adaptive partitioning of mass and time since accretion
- Mixing components from smoothed metallicity curves

Conclusion

- 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