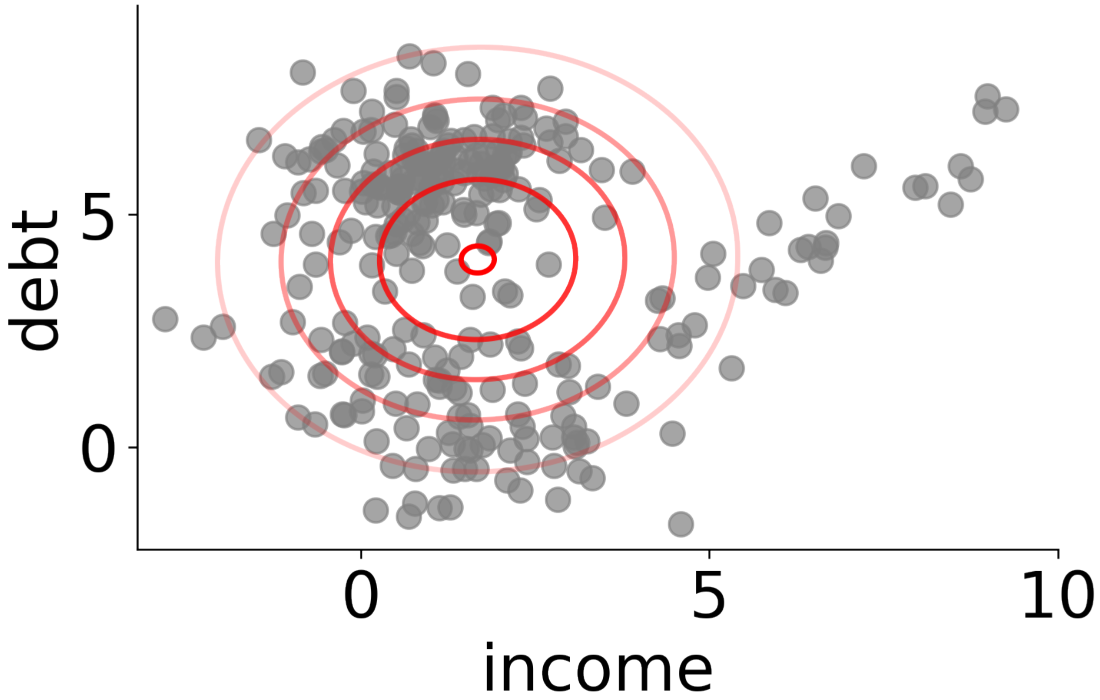
Probabilistic model of data

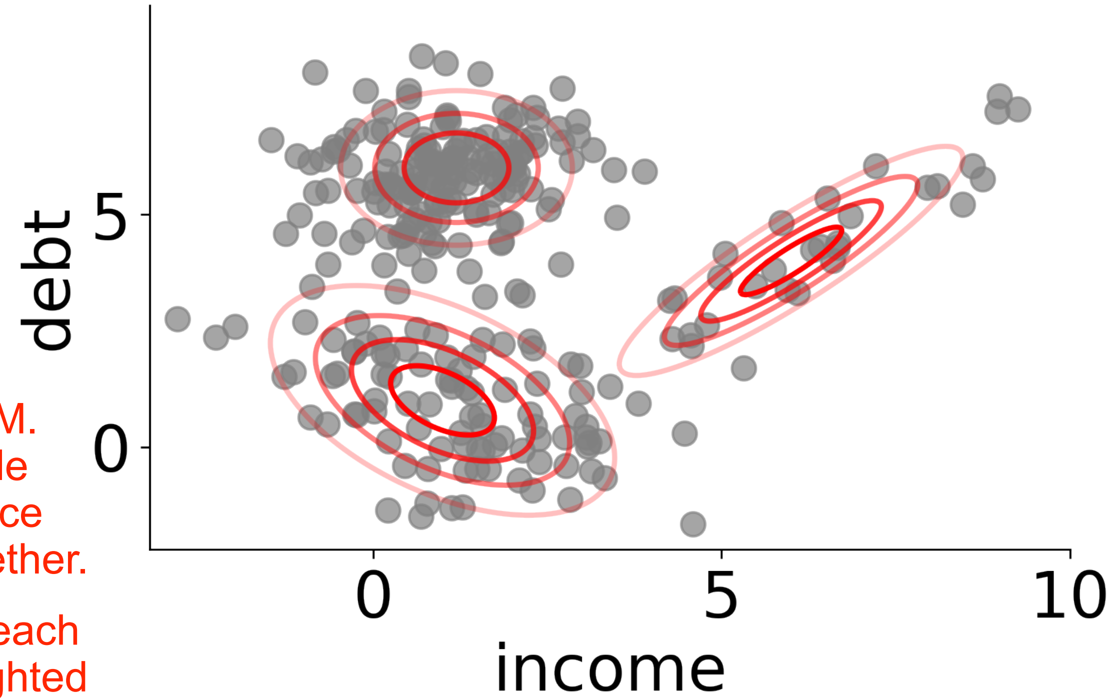


$$p(x \mid \theta) = \mathcal{N}(x \mid \mu, \Sigma)$$

$$\theta = \{\mu, \Sigma\}$$

we learned how to fit a gaussian into a data point.
we can see, the use of a single gaussian random variable is quite limiting - we are modelling everything within one big circle.
It can't really account for multiple modes, and thus clusters.

Gaussian Mixture Model (GMM)



This is the GMM. We use multiple gaussians, hence 'mixing them' together.

The density of each point is the weighted sum of the three gaussian densitieis.

$$p(x \mid \theta) = \pi_1 \mathcal{N}(x|\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(x|\mu_2, \Sigma_2) + \pi_3 \mathcal{N}(x|\mu_3, \Sigma_3)$$

$$\theta = \{\pi_1, \pi_2, \pi_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3\}$$

pi are the normalization constants that sum up to 1 to make an actual probability distribution.

When we succeed in fitting this model, we can ask for any data point: Which gaussian did it came from?

GMM vs Guassian

	Gaussian	GMM
Flexibility		
# of parameters		
		GMM has more parameters than Gaussian.
Parameters	μ, Σ	$\{\pi_1, \pi_2, \pi_3\}$ $\{\mu_1, \mu_2, \mu_3\}$ $\{\Sigma_1, \Sigma_2, \Sigma_3\}$

Training GMM

Originally, it is:
max_\theta P(X|theta).
However, x_i x_j are independent.
So we can form it as product of liklihood of independent objects.

Remember that the theta is the hyperparameters.

$$\max_{\theta}. \quad \prod_{i=1}^{N} p(x_i \mid \theta) = \prod_{i=1}^{N} (\pi_1 \mathcal{N}(x_i \mid \mu_1, \Sigma_1) + \ldots)$$

Maximise the likelihood of the density of the dataset, given the parameters.

$$\text{subject to} \quad \pi_1 + \pi_2 + \pi_3 = 1; \; \pi_k \geq 0; \; k = 1, 2, 3.$$

$$\Sigma_k \succ 0;$$

^^ The covariance matrices cannot be arbitrary.
The set of valid covariance matrices is something called positive semi-definite matrices.
This isn't the important part right now. The important thing to take note is that it is a hard constraint to follow. So we can't really use gradient descent.

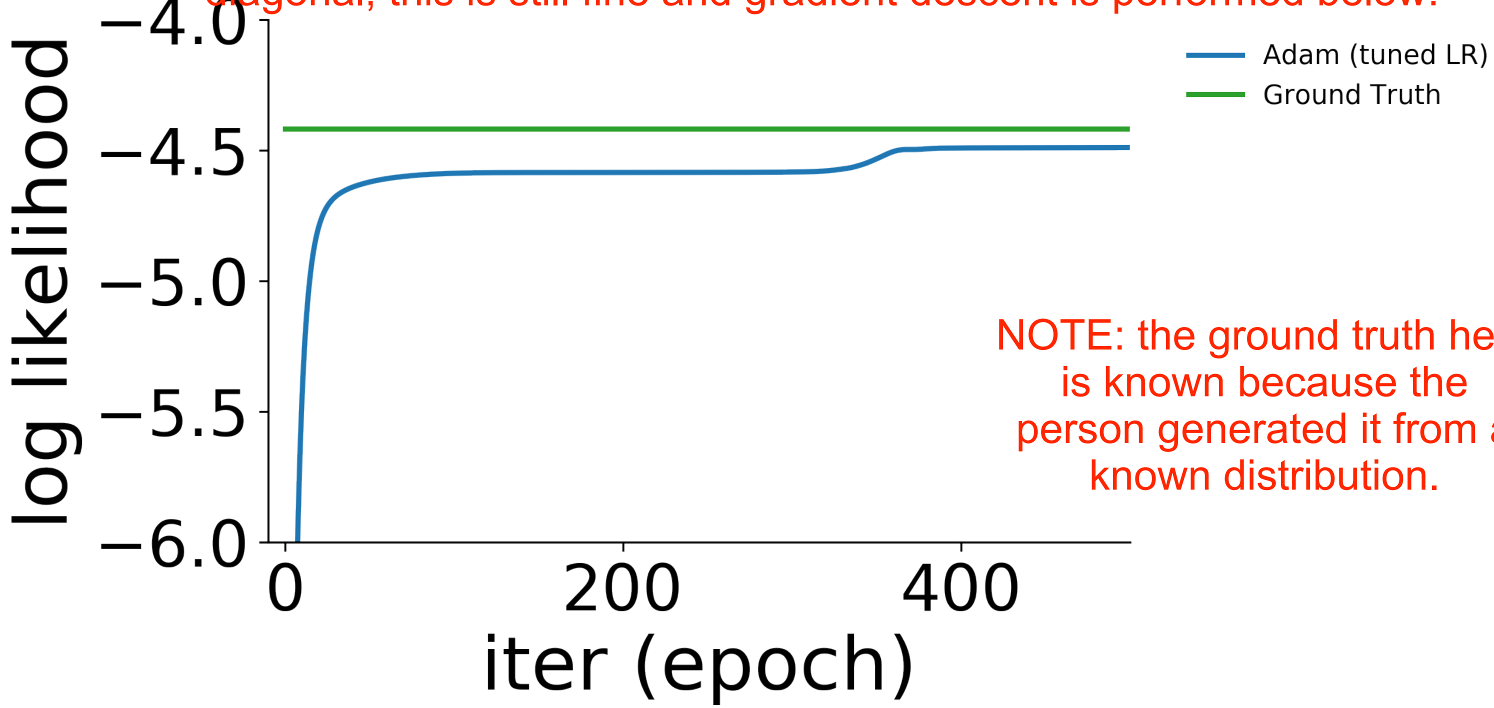
If you are curious, a matrix M is positive semidefinite IFF
1. M is symmetric (i.e. M^T = M)
2. v^T Mv >= 0 for all v in V, where M in L(V).

Training GMM

$$\max_{\theta} \quad \prod_{i=1}^{N} p(x_i \mid \theta) = \prod_{i=1}^{N} \left(\pi_1 \mathcal{N}(x_i \mid \mu_1, \Sigma_1) + \ldots\right)$$

$$\text{subject to} \quad \pi_1 + \pi_2 + \pi_3 = 1;\; \pi_k \geq 0;\; k = 1, 2, 3.$$

If we use an easier constraint, e.g. that the covariance matrices have to be diagonal, this is still fine and gradient descent is performed below:



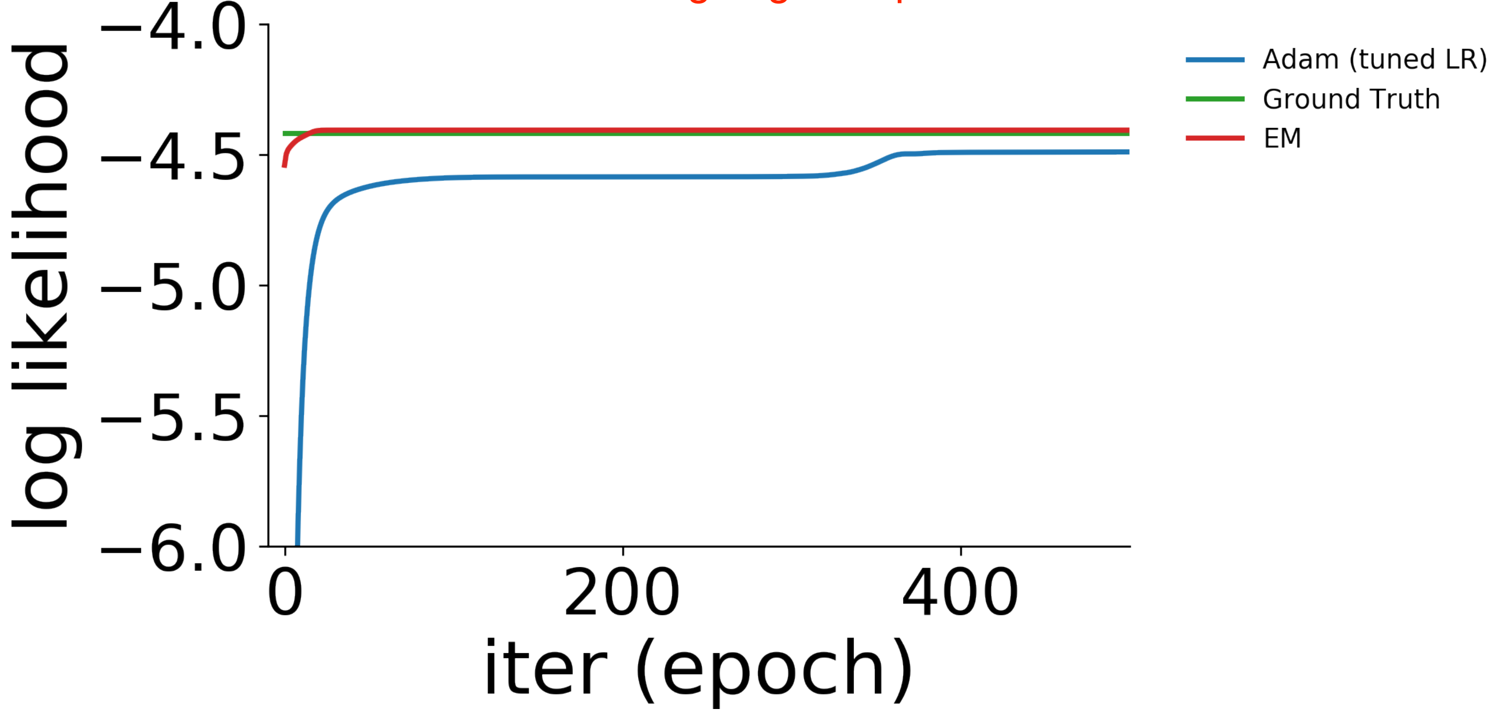
NOTE: the ground truth here is known because the person generated it from a known distribution.

Training GMM

$$\max_{\theta} \quad \prod_{i=1}^{N} p(x_i \mid \theta) = \prod_{i=1}^{N} \left(\pi_1 \mathcal{N}(x_i \mid \mu_1, \Sigma_1) + \ldots\right)$$

$$\text{subject to} \quad \pi_1 + \pi_2 + \pi_3 = 1; \; \pi_k \geq 0; \; k = 1, 2, 3.$$

But, a better way is to use EM. More efficient.
This is what we're going to explore now.



Summary

- Gaussian Mixture Model is a flexible probability distribution
- It is hard to fit (train) with SGD