Latent Variable Models

Saturday, June 23, 2018 12:11 AM

lvm1

Latent Variable Models

&

Expectation Maximization

Week 2

- What is a latent variable, why do we need it, and how to use it
- Common latent variable models (clustering and dimensionality reduction)

Week 2

- What is a latent variable, why do we need it, and how to use it
- Common latent variable models (clustering and dimensionality reduction)
- How to train them with Expectation Maximization algorithm E-M.
- Extensions of Expectation Maximization such as handling missing data

Latent (hidden) variable is a variable that you never observe

	High school grade
John	4.0
Helen	3.7
Jack	3.2
Emma	2.9

	High school grade	University grade
John	4.0	4.0
Helen	3.7	3.6
Jack	3.2	N/A
Emma	2.9	3.2

	High school grade	University grade	IQ score
John	4.0	4.0	120
Helen	3.7	3.6	N/A
Jack	3.2	N/A	112
Emma	2.9	3.2	N/A

	High school grade	University grade	IQ score	Phone Interview	Onsite interview
John	4.0	4.0	120	3/4	?
Helen	3.7	3.6	N/A	4/4	?
Jack	3.2	N/A	112	2/4	?
Emma	2.9	3.2	N/A	3/4	?

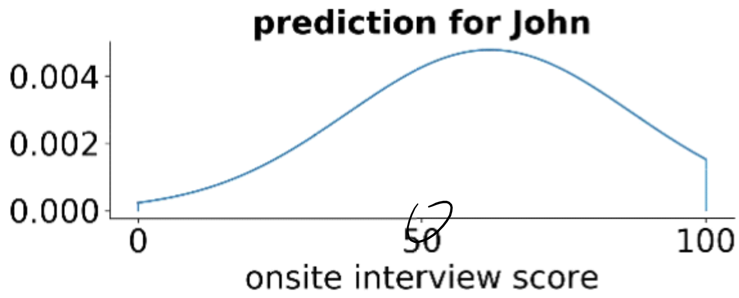
	High school grade	University grade	IQ score	Phone Interview	Onsite interview
Sophia	3.5	3.6	N/A	4/4	85/100
…					

	High school grade	University grade	IQ score	Phone Interview	Onsite interview
John	4.0	4.0	120	3/4	?
Helen	3.7	3.6	N/A	4/4	?
Jack	3.2	N/A	112	2/4	?
Emma	2.9	3.2	N/A	3/4	?

	High school grade	University grade	IQ score	Phone Interview	Onsite interview
Sophia	3.5	3.6	N/A	4/4	85/100
…					

missing values

	High school grade	University grade	IQ score	Phone Interview	Onsite interview
John	4.0	4.0	120	3/4	?
Helen	3.7	3.6	N/A	4/4	?
Jack	3.2	N/A	112	2/4	?
Emma	2.9	3.2	N/A	3/4	?



if 50 (middle)

medn — med-level

— unsure.

Probabilistic model

Probabilistic model

most flexible

least structural

Assign probability for each possible combination

prob. as parameters

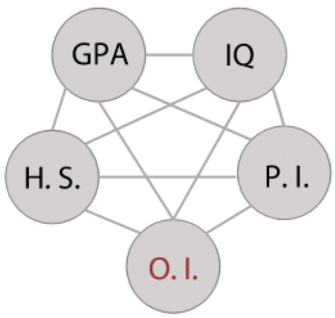
can use parametric model

Probabilistic model

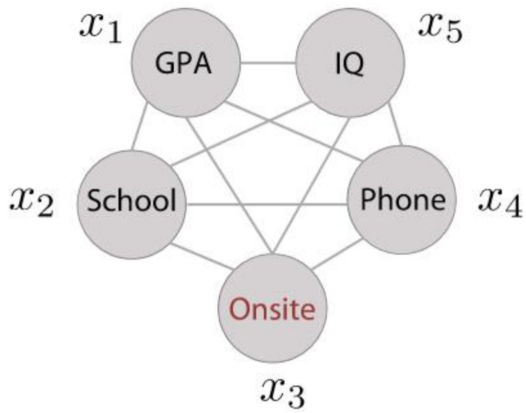
High	GPA	IQ	Phone	Onsite	Probability

Probabilistic model

High school	GPA	IQ	Phone Interview	Onsite Interview	Probability
1.0	1.0	1	0/4	1/100	0.001
1.0	1.0	1	0/4	2/100	0.0023
…	…	…	…		
4.0	4.0	180	4/4	100	0.000001



Probabilistic model



5 per var.

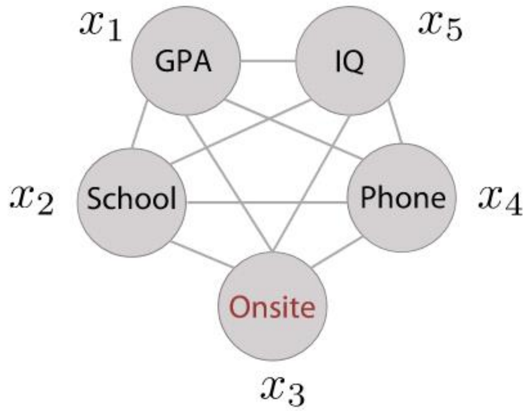
$$p(x_1, x_2, x_3, x_4, x_5) = \frac{exp(-w^\intercal x)}{Z}$$

reduce model complexity by a lot

Need Sum of all possible combination

Problem Training intractable

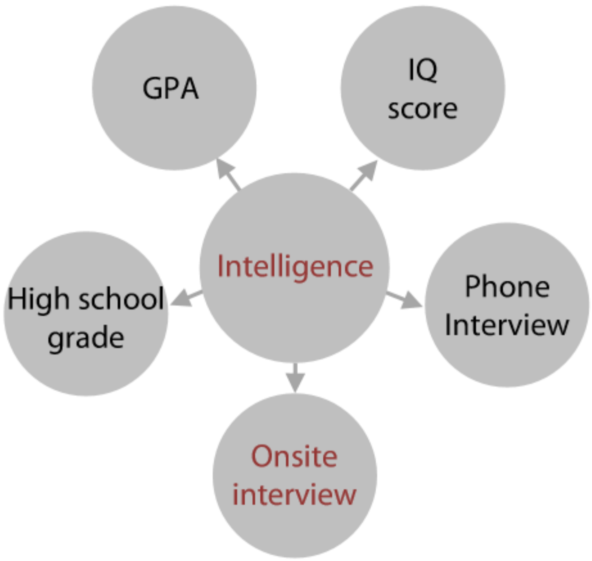
Probabilistic model



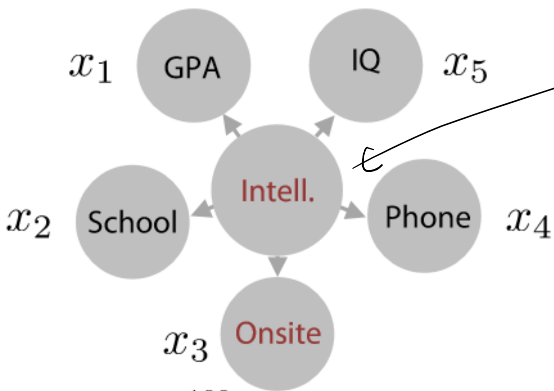
$$p(x_1, x_2, x_3, x_4, x_5) = \frac{exp(-w^{\intercal}x)}{Z}$$

→ sum of all possible configuration! Gigantic sum.

Probabilistic model



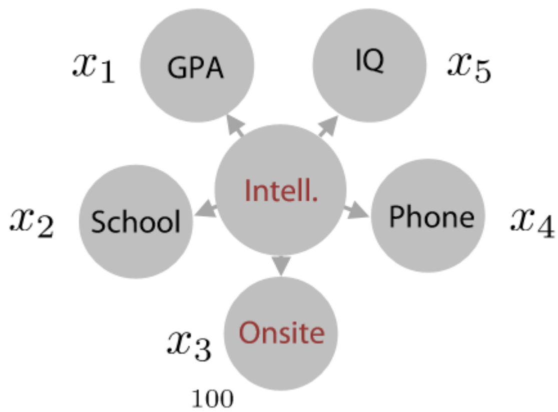
Probabilistic model



Add.
Reduce model complexity

$$p(x_1, x_2, x_3, x_4, x_5) = \sum_{I=1}^{100} p(x_1, x_2, x_3, x_4, x_5 \mid \underline{I})\, p(\underline{I})$$

Probabilistic model



reduce model complexity without reducing flexibility.

$$p(x_1, x_2, x_3, x_4, x_5) = \sum_{I=1}^{100} p(x_1, x_2, x_3, x_4, x_5 \mid I)\, p(I) =$$

Factorize into 5 small tables

$$\sum_{I=1}^{100} \underline{p(x_1 \mid I)} \dots \underline{p(x_5 \mid I)}\, p(I)$$

Do you think it is always a good idea to introduce latent variables?

- [x] No, sometimes adding latent variables restrict your model too much

Correct

Correct! If for example, a student is doing 2 tests in the same day, it doesn't make sense to assume that these two grades are caused only by his intelligence and doesn't influence each other directly. Even if we know that he is very smart, if he failed the first test he is more likely to fail the second one because he may have a headache or maybe he didn't have time to prepare the day before.

- [ ] No, if we don't have a variable in the training dataset we cannot add it as a latent variable

Un-selected is correct

- [x] No, sometimes there is no need for them

Correct

Correct! For example fitting a dataset into a Gaussian distribution is easy enough. By

Latent variable models

No, sometimes there is no need for them

Correct

Correct! For example fitting a dataset into a Gaussian distribution is easy enough. By

Latent variable models

Pros:

- Simpler models (less edges)

Latent variable models

Pros:

- Simpler models (less edges)
- Fewer parameters

Latent variable models

Pros:

- Simpler models (less edges)
- Fewer parameters
- Latent variables are sometimes meaningful

Latent variable models

Pros:

- Simpler models (less edges)
- Fewer parameters
- Latent variables are sometimes meaningful

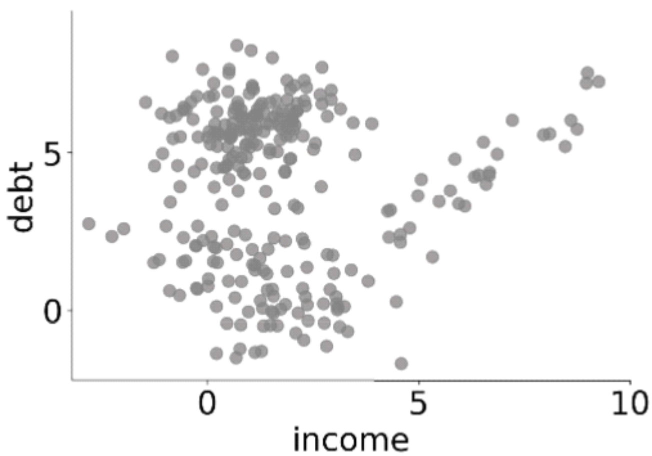
Cons:

- Harder to work with

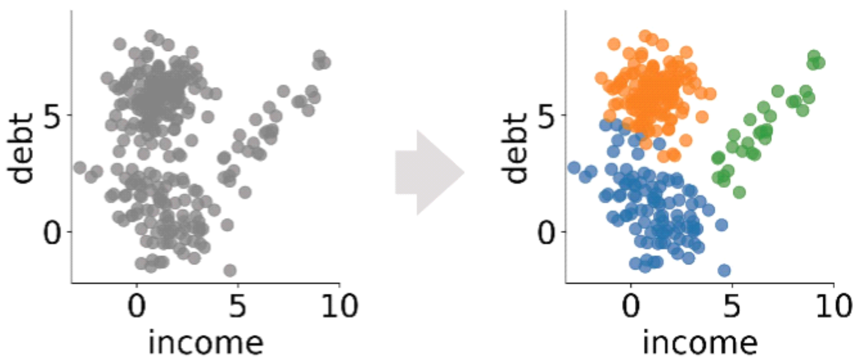
latent var for clustering

Probabilistic clustering

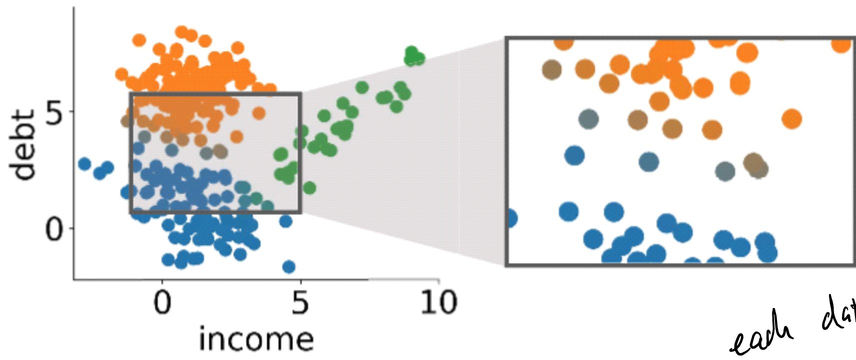
Clustering



Hard clustering



Soft clustering



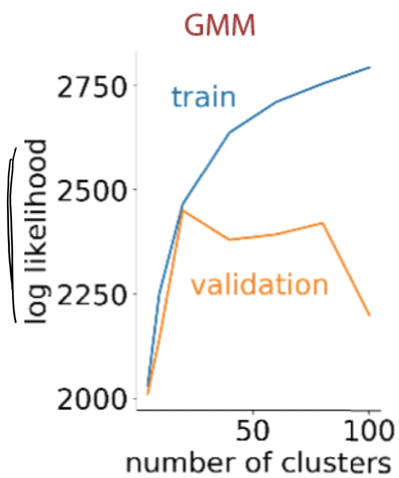
$$p(\text{cluster idx} \mid x)$$

instead of

$$\text{cluster idx} = f(x)$$

each data point assigned prob. distribution over clusters.

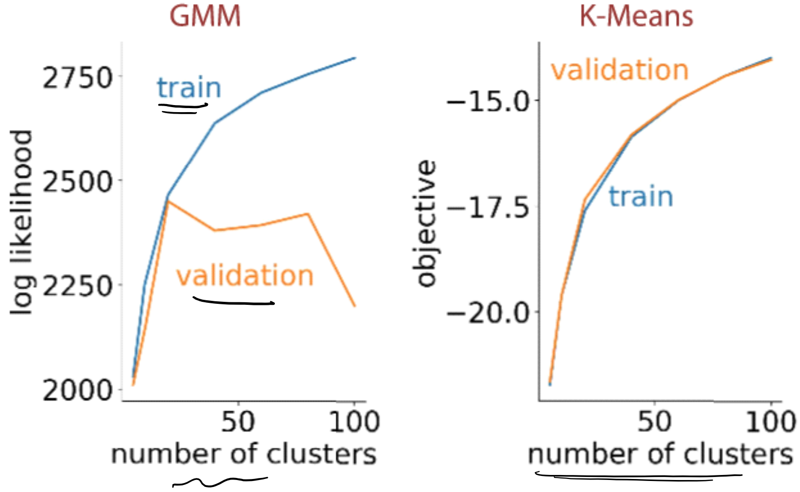
Hyperparameter tuning



# of cluster

Hyperparameter tuning

need validation, because perfect performance for in-sample if one sample one class.



First Reason why considering probabilistic approach to clustering

Generating new data points

Interesting

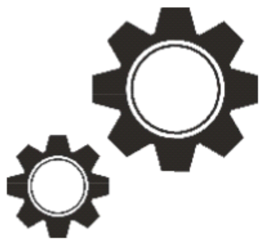
Junbo Zhao, https://arxiv.org/pdf/1609.03126.pdf

Generate fake celebrity faces-
~~by~~ by probabilistic model for clustering

Summary

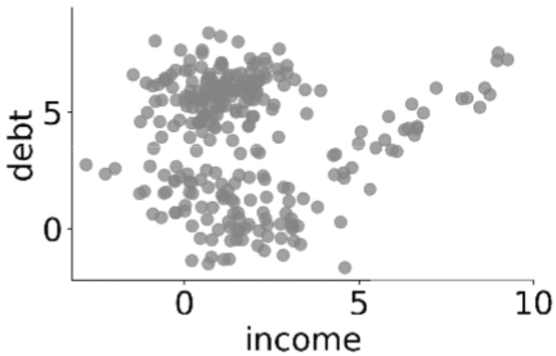
Want to cluster data in a soft way

- Allows to tune hyper parameters
- Generative model of the data

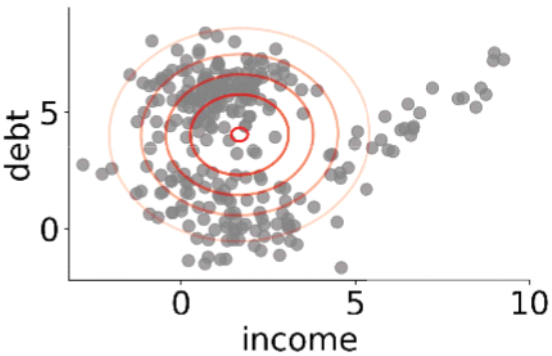


gmm1

Probabilistic model of data

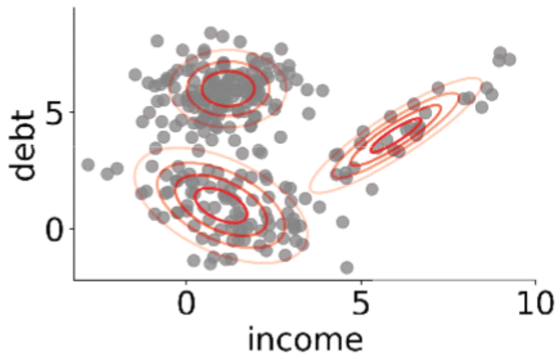


Probabilistic model of data

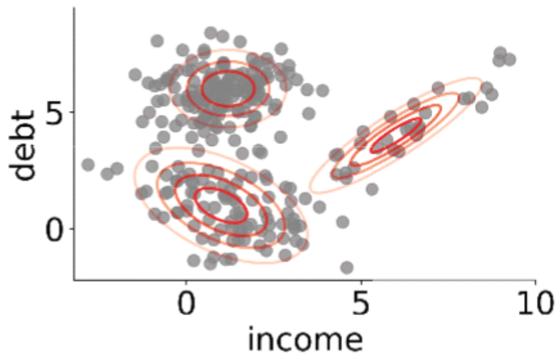


$$p(x \mid \theta) = \mathcal{N}(x \mid \mu, \Sigma)$$
$$\theta = \{\mu, \Sigma\}$$

Gaussian Mixture Model (GMM)

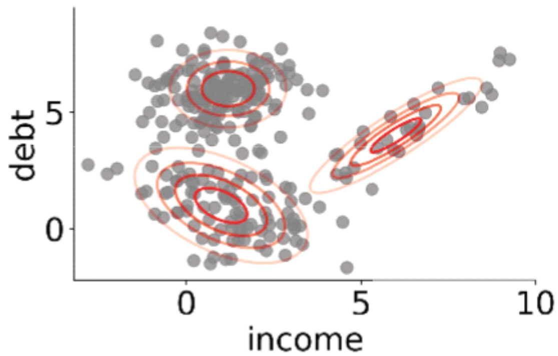


Gaussian Mixture Model (GMM)



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

Gaussian Mixture Model (GMM)



$$p(x \mid \theta) = \pi_1 \mathcal{N}(x|\mu_1, \overset{G1}{\Sigma_1}) + \pi_2 \mathcal{N}(x|\mu_2, \Sigma_2) + \pi_3 \mathcal{N}(x|\mu_3, \Sigma_3)$$

weighted sum of 3 Gaussian density.

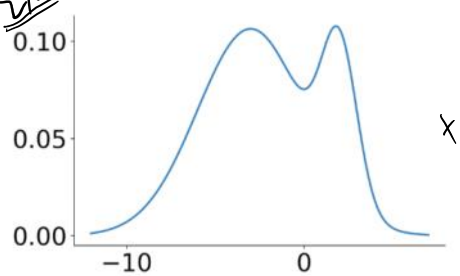
$$\theta = \{\pi_1, \pi_2, \pi_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3\}$$

Gaussian Mixture Model (GMM)

	Gaussian	GMM
Flexibility		

GMM vs Guassian

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



What are the parameters of the two components of 1-dimensional Gaussian Mixture Model which density is plotted above?

Recall that the density of the mixture is defined as follows

$$p(x \mid \pi_1, \pi_2, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = \pi_1 \mathcal{N}(x \mid \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x \mid \mu_2, \sigma_2^2)$$

X
- $\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = -3, \mu_2 = 2, \sigma_1 = 3, \sigma_2 = 1$

  This should not be selected
  Almost correct, but note that if $\pi_1 = \pi_2 = 0.5$, the right Gaussian would be much higher than the lest Gaussian because of the difference in variances.

- $\pi_1 = 0.8, \pi_2 = 0.2, \mu_1 = -3, \mu_2 = 2, \sigma_1 = 1, \sigma_2 = 3$
- $\pi_1 = 0.5, \pi_2 = 0.5, \mu_1 = -3, \mu_2 = 2, \sigma_1 = 1, \sigma_2 = 3$
- $\pi_1 = 0.8, \pi_2 = 0.2, \mu_1 = -3, \mu_2 = 2, \sigma_1 = 3, \sigma_2 = 1$

Training GMM

$$\max_{\theta}. \quad p(X \mid \theta)$$

Training GMM

$$\max_{\theta}. \quad p(X \mid \theta) = \prod_{i=1}^{N} p(x_i \mid \theta)$$

Training GMM

N data points independent w/ parameters

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

Training GMM

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

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

Training GMM

这些是probability吗?

Continuous var 的 mixture 也和 discrete 一样 加起来是1吗.

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

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

$$\Sigma_k \succ 0;$$

$\longrightarrow$ matrix always semi-definite

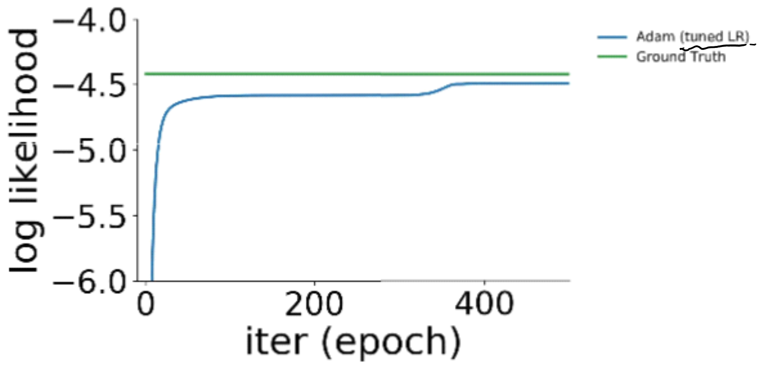
Hard to use SGD for this

HC Why hard?

Training GMM

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

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



Assume diagonal Σ_k.

can use SGD ?)

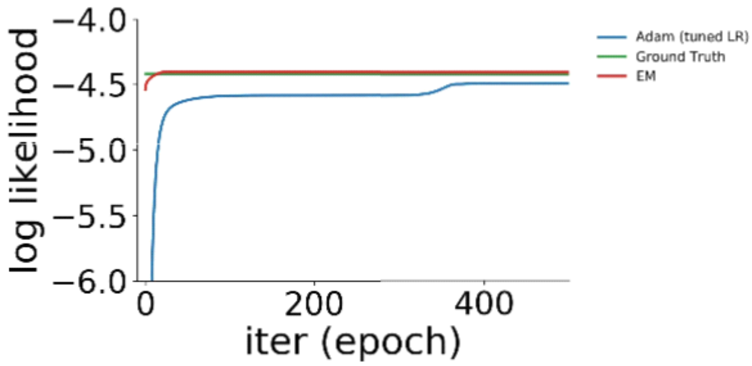
Hard

Even get rid of some constraints

Training GMM

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

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



EM works
So much better.
[even better than ground truth]

Not use SGD . can not ~~incorporate~~ use constraints

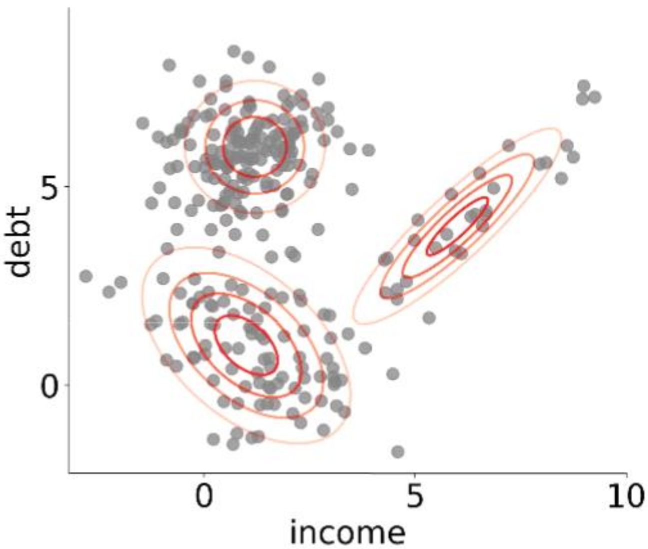
. Slow / less efficient

Summary

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

gmm2

Gaussian Mixture Model (GMM)



Introducing latent variable

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

Introducing latent variable

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

latent variable

has prior distribution π

$$p(t=c \mid \theta) = \pi_c$$

$$p(x \mid t=c, \theta) = \mathcal{N}(x \mid \mu_c, \Sigma_c).$$

Introducing latent variable

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

$$p(t = c \mid \theta) = \pi_c$$

Introducing latent variable

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

$$p(t = c \mid \theta) = \pi_c$$

$$p(x \mid t = c, \theta) = \mathcal{N}(x \mid \mu_c, \Sigma_c)$$

Introducing latent variable

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

$$p(t = c \mid \theta) = \pi_c$$

$$p(x \mid t = c, \theta) = \mathcal{N}(x \mid \mu_c, \Sigma_c)$$

$$p(x \mid \theta) = \underline{\sum_{c=1}^{3} p(x \mid t = c, \theta) p(t = c \mid \theta)}$$

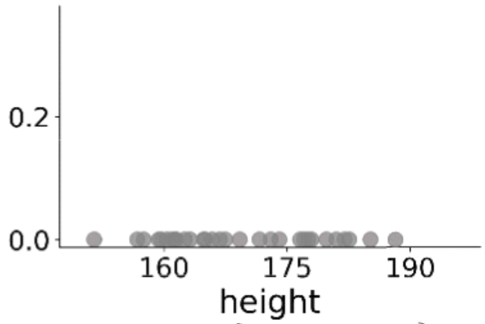
Marginalize out t.

Introduce latent variable

latent variable introduced for clustering!

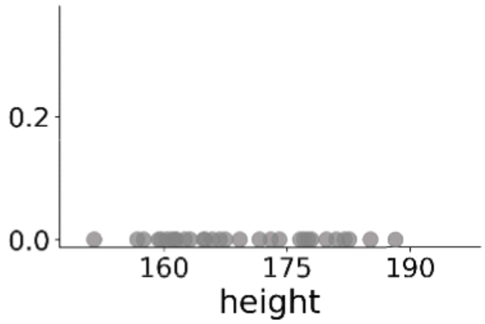
exactly same form as the version before introducing latent variable.

Expectation Maximization



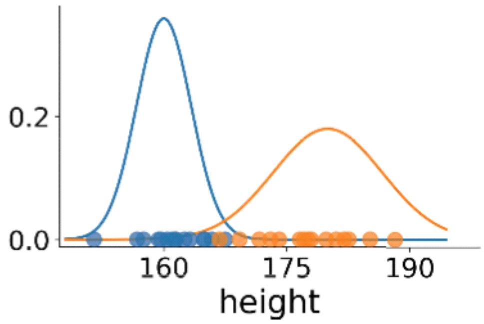
Dataset: $\{x_1, \ldots, x_N\}$

Expectation Maximization



How to estimate parameter θ?

Expectation Maximization



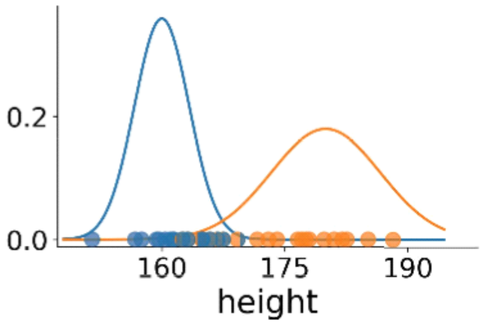
How to estimate parameter θ?

If sources t are known, easy:

$$p(x \mid t = 1, \theta) = \mathcal{N}(x \mid \mu_1, \sigma_1^2)$$

$$\mu_1 = \frac{\sum_{\text{blue } i} x_i}{\#\text{ of blue points}} \qquad \sigma_1^2 = \frac{\sum_{\text{blue } i} (x_i - \mu_1)^2}{\#\text{ of blue points}}$$

Expectation Maximization



How to estimate parameter θ?

If sources t are known, easy:

variance

$p(t_i | x_i, \theta)$

want $\sum_{(t_i=1)} x \, p(x_i | t_i = 1, \theta)$

$$\mu_1 = \frac{\sum_i p(t_i = 1 \mid x_i, \theta)\, x_i}{\sum_i p(t_i = 1 \mid x_i, \theta)} \qquad \sigma_1^2 = \frac{\sum_i p(t_i = 1 \mid x_i, \theta)\, (x_i - \mu_1)^2}{\sum_i p(t_i = 1 \mid x_i, \theta)}$$

ex

If we know for each point from which Gaussian (cluster) it was generated, then to compute the mean parameter of the first Gaussian we may use the following expression:

$$\mu_1 = \frac{\sum_{i \text{ from 1st Gaussian}} x_i}{\text{\# points from the 1st Gaussian}}$$

So we essentially find the sample mean across all the points from the first Gaussian.

What if now we know about each point its posterior distribution on the latent variable t_i: $p(t_i \mid x_i, \theta)$, that is we don't know exactly from which cluster this point came, but we have a distribution. How can we compute the mean parameter of the first Gaussian in this case?

If you don't know how to compute the answer, don't worry and just give your best guess!

- $\mu_1 = \frac{\sum_i x_i}{N}$
- $\mu_1 = \frac{\sum_i p(t_i=1|x_i,\theta) x_i}{N}$
- $\mu_1 = \frac{\sum_i p(t_i=1|x_i,\theta) x_i}{\sum_i p(t_i=1|x_i,\theta)}$

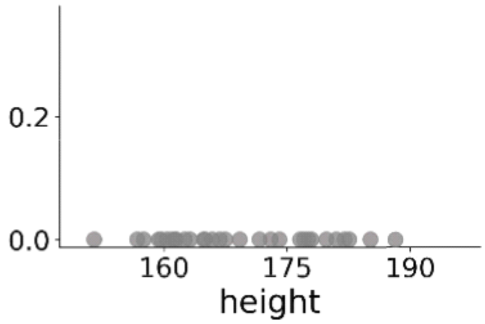
Correct

Expectation Maximization

$\mu_1 = \frac{\sum_i p(t_i=1|x_i,\theta)x_i}{\sum_i p(t_i=1|x_i,\theta)}$

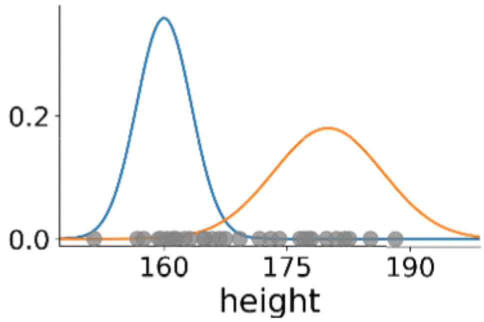
Correct

Expectation Maximization



What if we don't know the sources?

Expectation Maximization

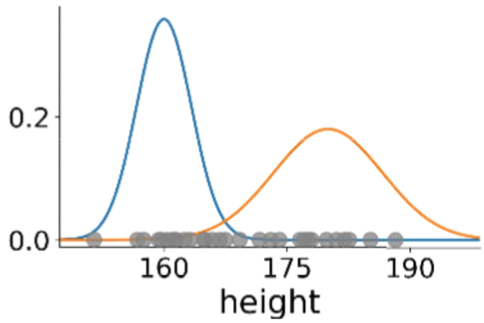


What if we don't know the sources?

Given: $p(x \mid t = 1, \theta) = \mathcal{N}(-2, 1)$

Find: $p(t = 1 \mid x, \theta)$

Expectation Maximization



What if we don't know the sources?

If we know parameters θ, easy:

Σ

what's θ = μ, Σ

mean

$$p(t = 1 \mid x, \theta) = \frac{p(x \mid t = 1, \theta)\, p(t = 1 \mid \theta)}{Z}$$

Expectation Maximization

Chicken and egg problem

- Need Gaussian parameters to estimate sources
- Need sources to estimate Gaussian parameters

Expectation Maximization

Chicken and egg problem

- Need Gaussian parameters to estimate sources
- Need sources to estimate Gaussian parameters

EM algorithm

1. Start with 2 randomly placed Gaussians parameters θ
2. Until convergence repeat:
   a) For each point compute $p(t = c \mid x_i, \theta)$: does x_i look like it came from cluster c?
   b) Update Gaussian parameters θ to fit points assigned to them

gmm3

Expectation Maximization

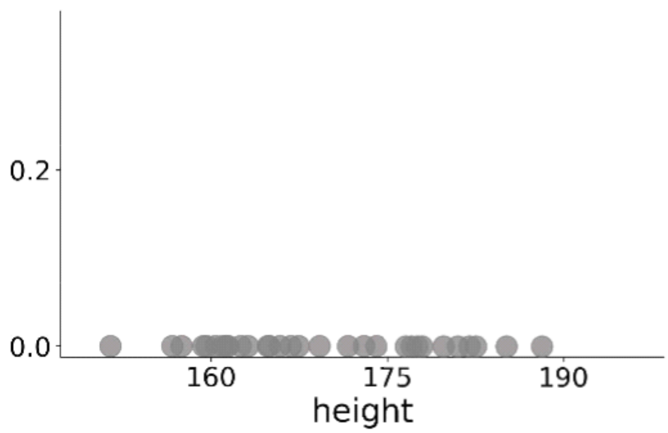
Chicken and egg problem

- Need Gaussian parameters to estimate sources
- Need sources to estimate Gaussian parameters

EM algorithm

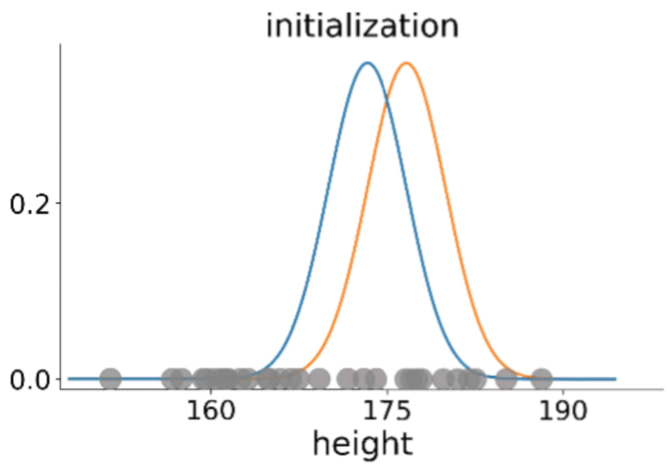
1. Start with 2 randomly placed Gaussians parameters θ
2. Until convergence repeat:
   a) For each point compute $p(t = c \mid x_i, \theta)$: does x_i look like it came from cluster c?
   b) Update Gaussian parameters θ to fit points assigned to them

GMM EM example

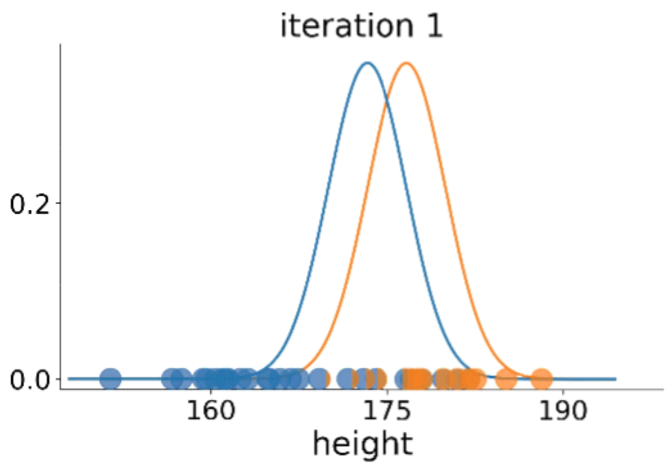


GMM EM example

Random initialization

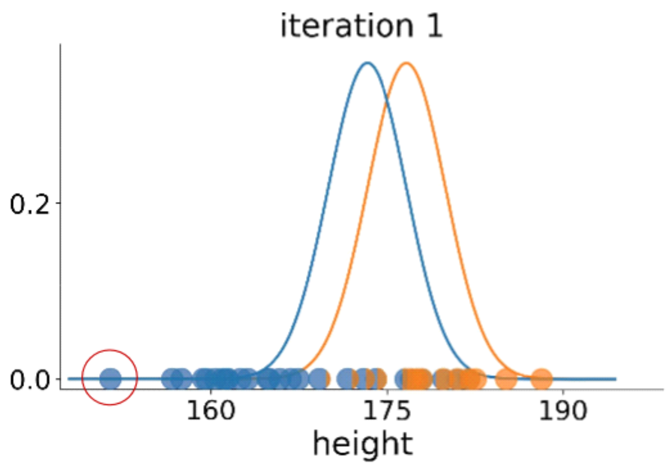


GMM EM example



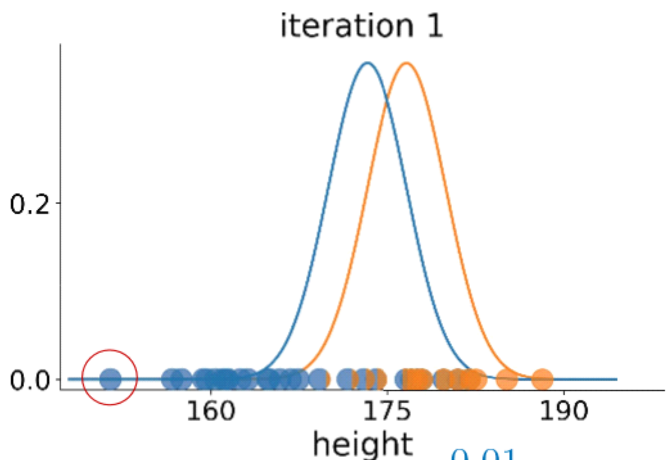
$$p(t_i = 1 \mid x_i, \theta)$$

GMM EM example



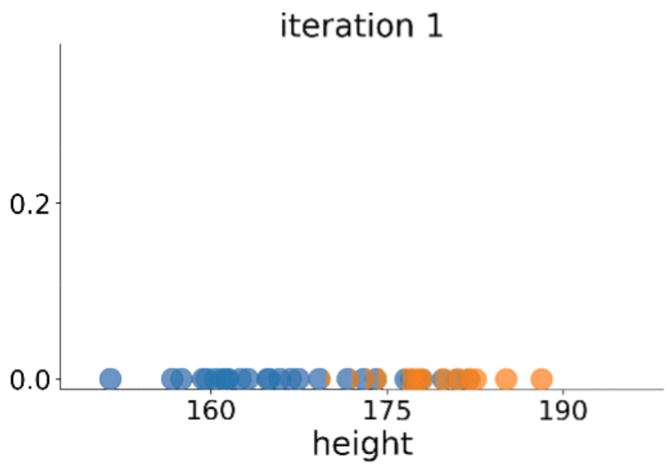
$$p(t_i = 1 \mid x_i, \theta)$$

GMM EM example



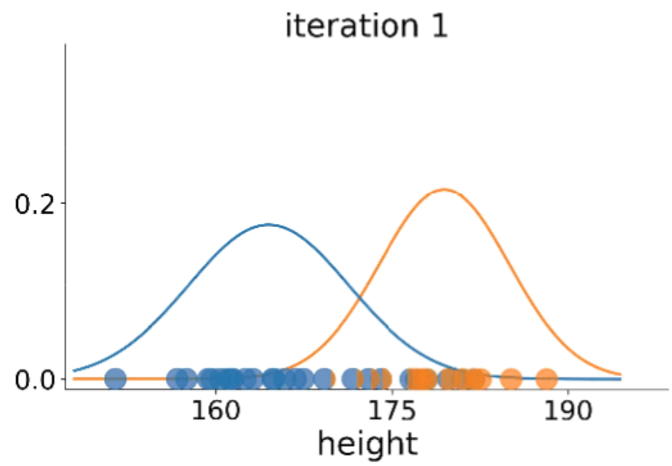
$$p(t_i = 1 \mid x_i, \theta) \approx \frac{0.01}{0.01 + 0.00002} \approx 1$$

GMM EM example

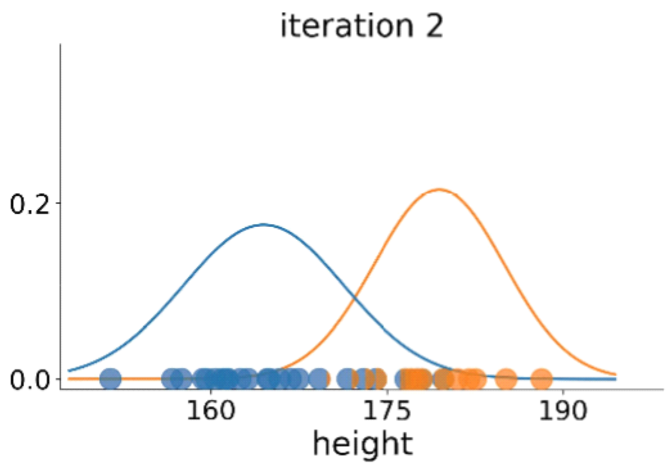


$$p(t_i = 1 \mid x_i, \theta)$$

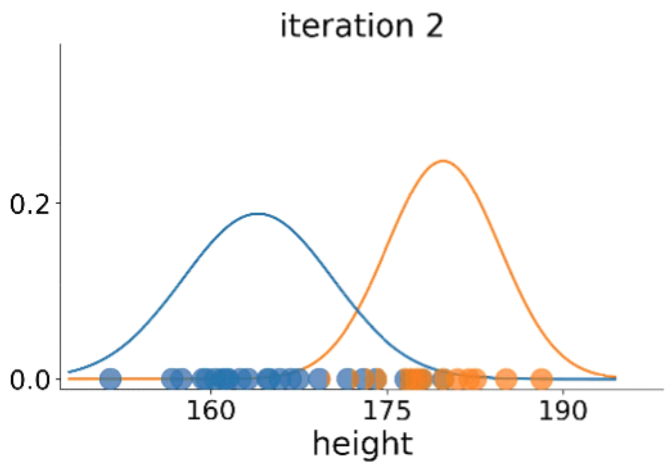
GMM EM example



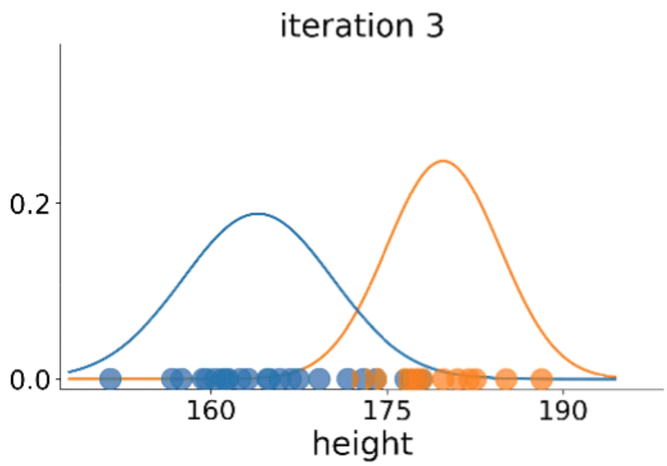
GMM EM example



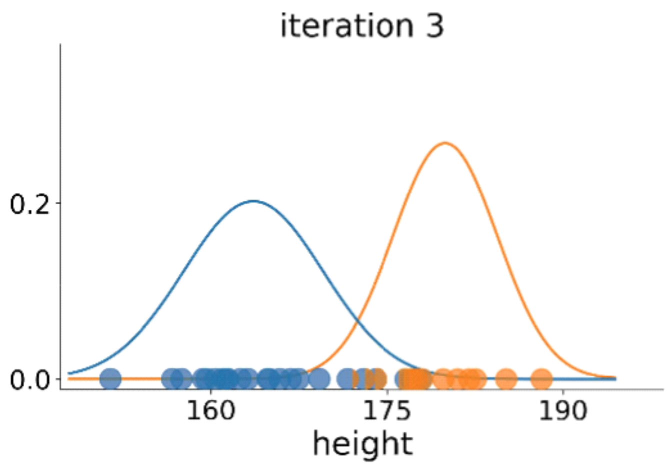
GMM EM example



GMM EM example



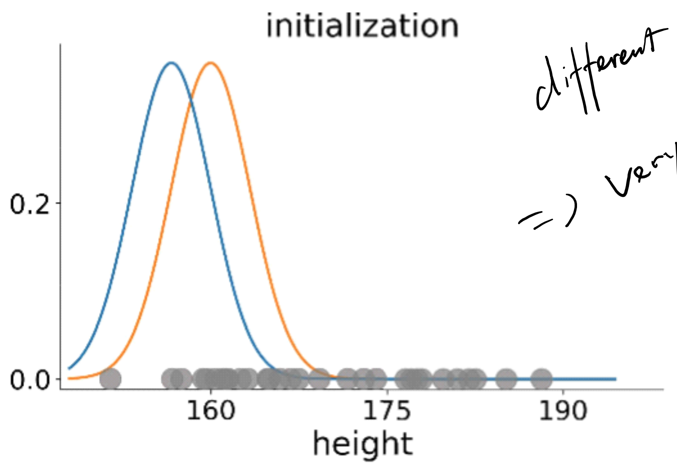
GMM EM example



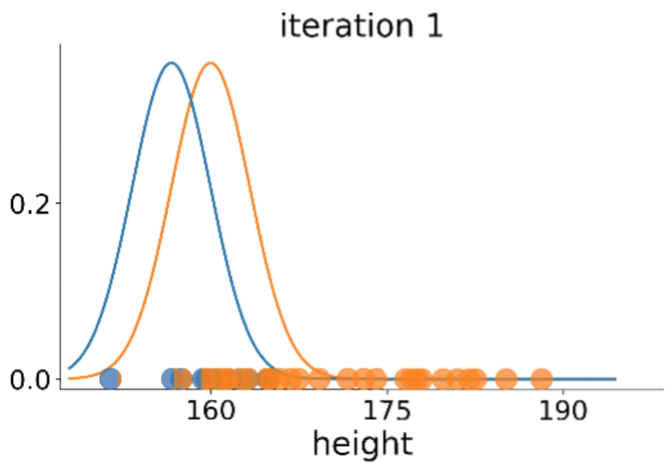
GMM EM local maximum example

GMM EM local maximum example

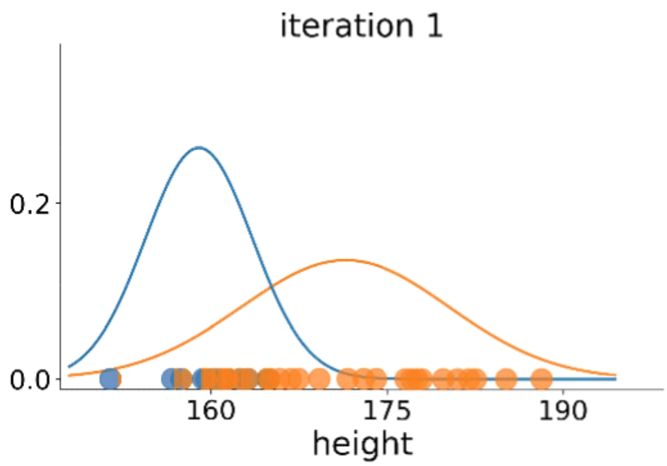
Same data
different initialization (to the left)
=) very different result!



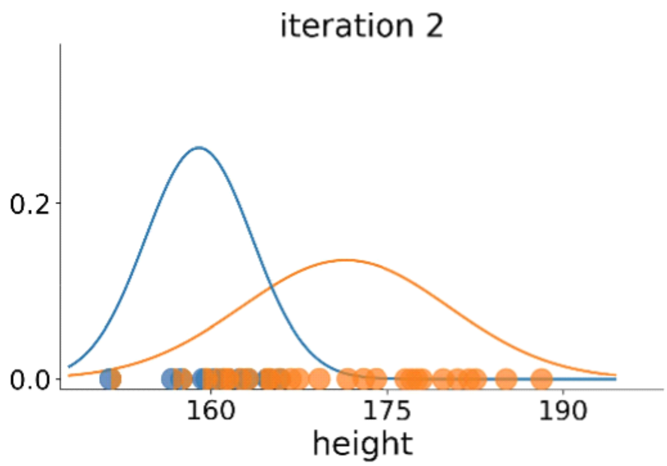
GMM EM local maximum example



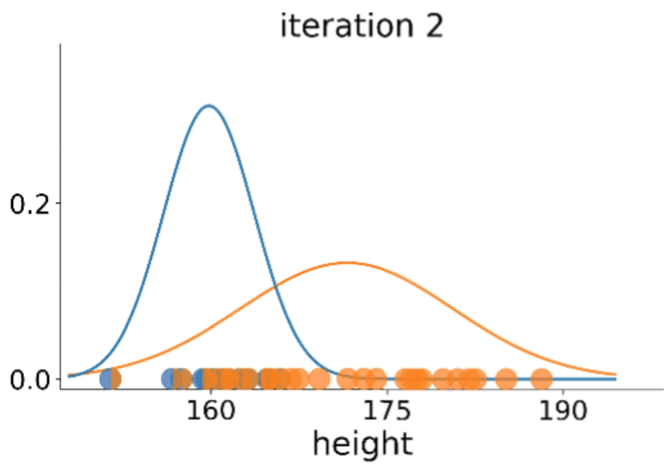
GMM EM local maximum example



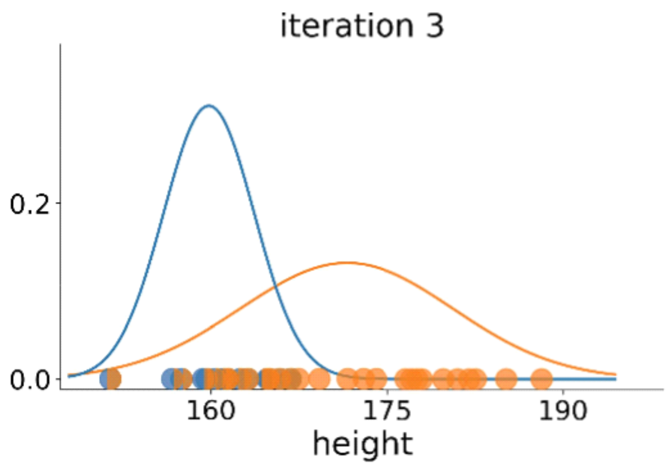
GMM EM local maximum example



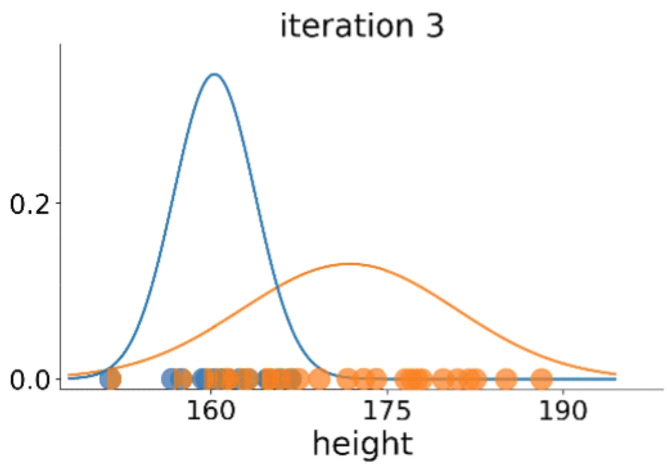
GMM EM local maximum example



GMM EM local maximum example



GMM EM local maximum example



Summary

- Gaussian Mixture Model is a flexible probabilistic approach to clustering problem
- Expectation Maximization algorithm can train GMM faster than Stochastic Gradient Descent and also handles complicated constraint
- Expectation Maximization suffers from local maxima (the exact solution is NP-hard)

Solution
start from
different initialization.

How can we choose the best run among several training attempts with different random initializations? Choose all answers that make sense.

- [ ] Choose the global maximum (while ignoring local maximums)

Un-selected is correct

- [x] Choose the one with the highest training log-likelihood

Correct
This is the standard way to deal with local maximums of any objective: among several runs choose the one that has the highest value of the objective.

- [x] Choose the one with the highest validation log-likelihood

Correct
This is a valid approach, although it feels a little bit weird: we are basically tunning the random seed on the validation set.