Stochastic Gradient Search and Stochastic Approximation for MLE Approximation

By Henry Bourne

Table of Contents

What's the MLE?

- Given a dataset, D, and parameters θ
- We define the likelihood as: $p(D|\theta)$
- Then the MLE is: $\hat{\theta} := \mathrm{argmax}_\theta \, p(D|\theta)$
- Why the MLE?
  - Generalizable (regression and classification)
  - MLE efficient (no consistent estimator has lower asymptotic error than MLE – if correct distribution being used)
  - Has approx. normal distribution (often) so with sample var. can easily compute confidence bounds and hypothesis tests

Analytical solution

- Sometimes we can directly find an analytical solution
- Start with the likelihood (assuming data iid):

$$L(\theta) := \prod_i p(x_i|\theta)$$

- Then we take the log (usually we use the negative of this)

$$log(L(\theta)) := \sum_i log(p(x_i|\theta))$$

- Then we take partial derivatives, set them to zero and solve

$$\frac{\partial log(L(\theta))}{\partial \theta^i} = 0$$

Some scenarios where we can't find analytical solutions:

- If likelihood function not differentiable
- If $\frac{\partial log(L(\theta))}{\partial \theta^i} = 0$ has no solutions
- If there are latent variables

Latent variables

- A variable that isn't (or can't) be directly observed
- The MLE assumes dataset is complete/fully observed
  - Ie. Assumes all variables relevant to the problem are present
- An Example:
  - Gaussian Mixture Model (GMM): For iid x $p(x_i) = \sum_{j=1,\ldots,K} w^{(j)} N_{x_i}(\mu^{(j)}, \sigma^{(j)})$
  - Here our model parameters are: $\theta = \begin{pmatrix} \boldsymbol{w} \\ \boldsymbol{\mu} \\ \boldsymbol{\sigma} \end{pmatrix}$
  - Where: $\sum_j w^{(j)} = 1$

Where's the latent variable?

- The mixture weights effectively are probabilities of a data point, x_i coming from their corresponding distributions
- Let our latent variable for observation i, z_i , be one-hot encoded, then

$$p(z_i^{(j)} = 1) = w^{(j)}$$

- To show how the latent variable is involved we can derive p(x) using our latent variable (ask me if you're interested!)

MLE for GMM

$$L(\theta) = \prod_{i=1}^{d} \sum_{j=1,\ldots,K} w^{(j)} N_{x_i}(\mu^{(j)}, \sigma^{(j)})$$

$$\log(L(\theta)) = \sum_{i=1}^{d} \log(\sum_{j=1,\ldots,K} w^{(j)} N_{x_i}(\mu^{(j)}, \sigma^{(j)}))$$

- Problems:
  - Difficult to reduce and therefore difficult to find analytical solutions

  - If one of the components of the GMM explains only a single point then the variance of the component can tend to 0, and L will then act like a delta function
    - So jumping straight to just maximizing the likelihood might not be such a good idea

GD for MLE

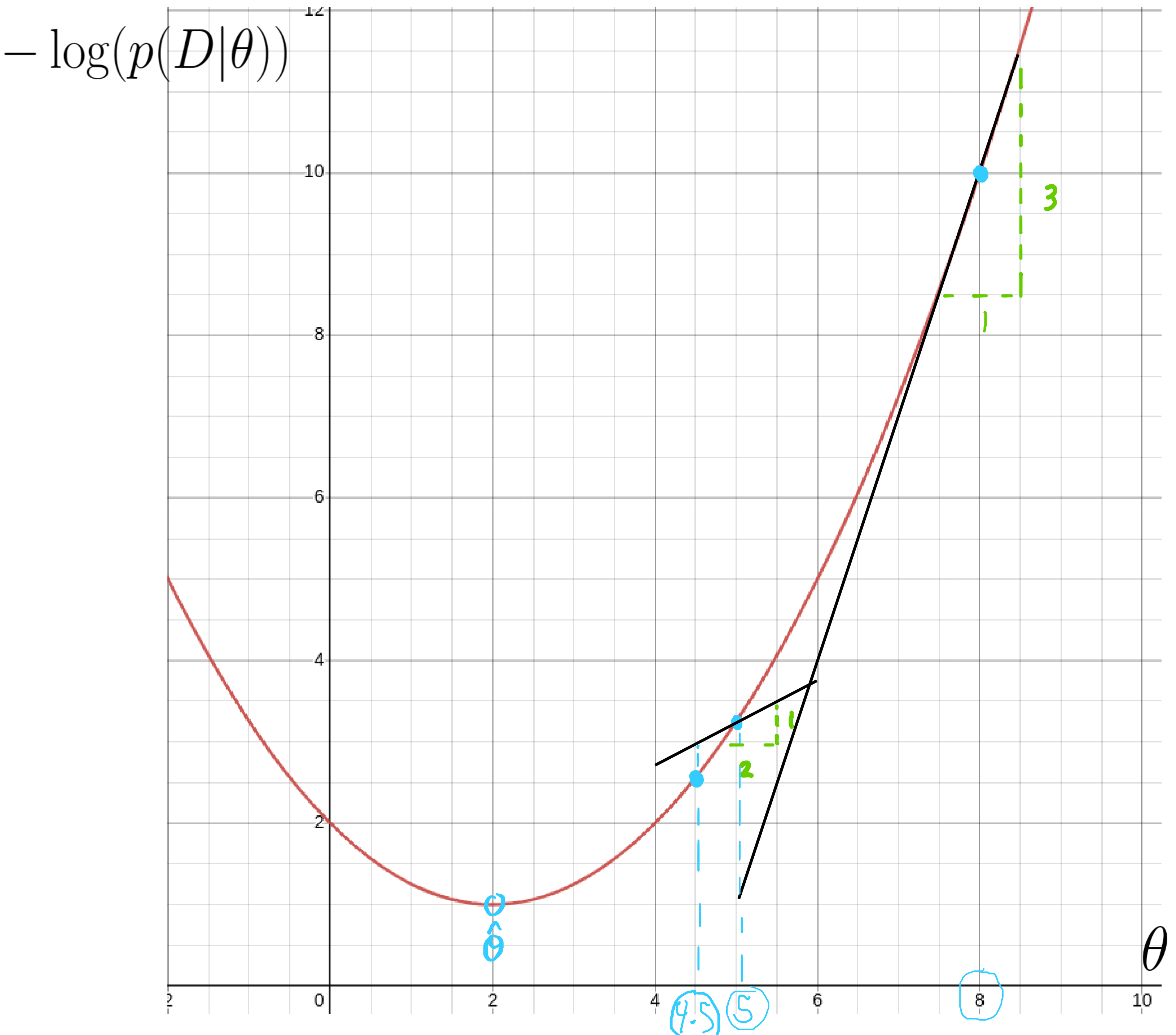
- One solution is Gradient Descent (GD)

1. Compute $\frac{\partial L}{\partial \theta^i}, \forall i$ (<u>Over all our data D</u>)
2. Let's say our current best guess is: $\hat{\theta}_t$
3. We can then find our next best guess: $\hat{\theta}_{t+1} = \hat{\theta}_t - \alpha \cdot \frac{\partial L}{\partial \theta}(\hat{\theta}_t)$

(Note: when working with GMMs we must also include a condition relating to the variance...)

Graphical representation of GD

- Use local knowledge of descent direction to guide us down the slope of the function
  $\alpha = 1$



However…

- Computing the gradients over all the data requires a large amount of compute

Stochastic GD (SGD) for MLE

- Idea is we use the help of randomness to reduce compute
- Process:
  - Sample random data-point (SGD) / subset of the data (minibatch GD)
  - Perform (one-step) GD on likelihood for the subset of data
  - Repeat
- Here we only have to compute the gradient over the data in our subset!
  - (Overall go over less data-points: as with GD need to use whole dataset for each iteration)

Why does stochastic/minibatch GD work?

- Assuming the samples we use are iid:
  - Gradient calculations on our samples = Unbiased estimators of the gradient of the whole dataset
  - ie. $\mathbb{E}(\hat{g}_{D' \subset D} - \hat{g}_D) = 0$
  - Means that the expected direction of the stochastic gradient is the same as the "full" gradient
- Bonus: stochasticity can help us escape local minima

Drawbacks of SGD

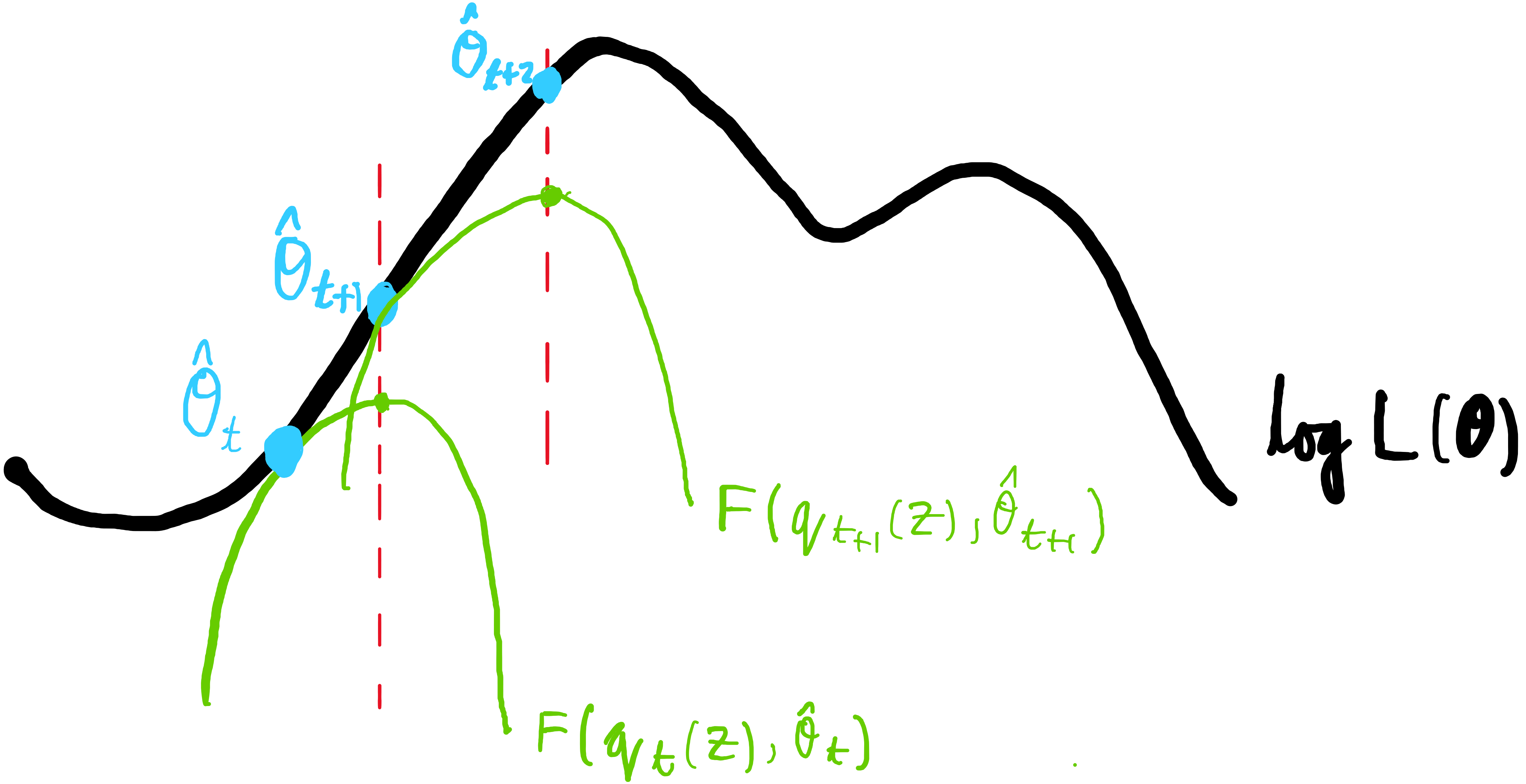
- We can use it even though we have latent variables!, however...
- We have to compute the first derivative of the likelihood
    - This can be intractable
    - Eg. Laplace distribution

The EM algorithm

- Doesn't need any derivatives! (… maybe?)
- Can be used with latent variables!
- More advantages?....

- Idea: find lower bound and raise it
  - Will continuously raise lower bound of the likelihood
- Now we have a lower bound, how do we raise it?
  - Problem: we have two arguments to maximize: $q(z), \theta$

Raising the lower bound



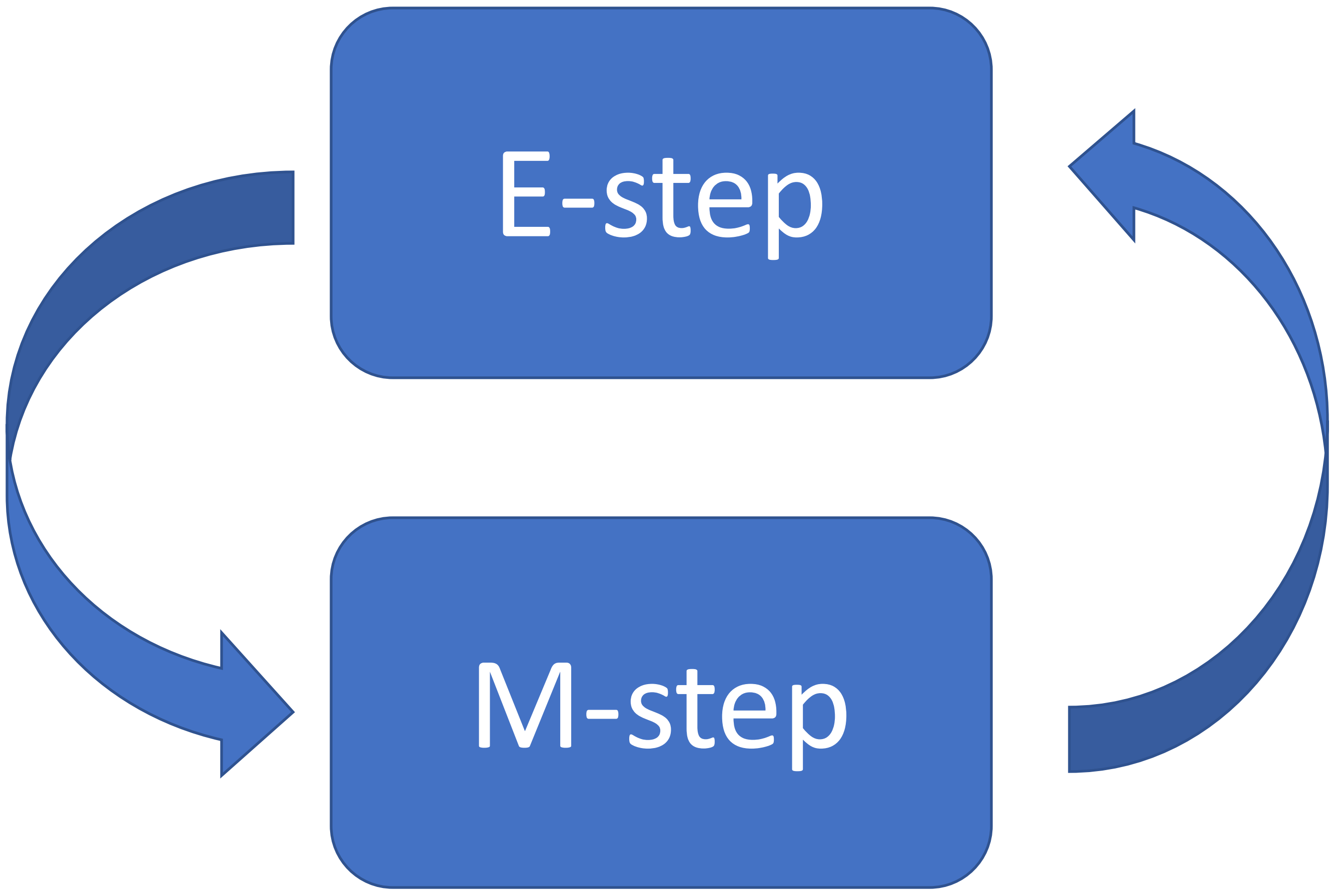
E-step

- Estimates missing or latent variables
- Let our "best guesses" be: $q_t(z), \hat{\theta}_t$
- We let $\hat{\theta}_t$ be fixed
- Want to maximize: $F(q_t(z), \hat{\theta}_t)$
    - Intuitively: this happens when we choose q s.t. F meets the likelihood
    - Mathematically:
        - Likelihood independent of $q(z)$
        - Recall, $F(q_t(z), \hat{\theta}_t) = \mathbb{E}(\log(L(\hat{\theta}_t))) - KL(q(z), p(z|y, \hat{\theta}_t))$
        - So maximum is equivalent to minimizing the KL divergence
- E-step: $q_{t+1}(z) = p(z|y, \hat{\theta}_t)$

M-step

- Now we fix $q_{t+1}(z)$
- And want to maximize: $F(q_{t+1}(z), \hat{\theta}_t)$
  - We can write F as:
    $$F(q_{t+1}(z), \hat{\theta}_t) = \int q_{t+1}(z) \log[p(y|z, \hat{\theta}_t) \cdot p(z|\hat{\theta}_t)] dz - \int q_{t+1}(z) \log[q_{t+1}(z)] dz$$
  - Note: second term doesn't depend on theta
  - So:
    $$\hat{\theta}_{t+1} = \text{argmax}_\theta \int q_{t+1}(z) \log[p(y|z, \theta) \cdot p(z|\theta)] dz$$
  - (Can find this analytically?)

The algorithm



(Note: How could we avoid problem with GMMs highlighted earlier?)

Advantages of the EM algorithm

- Already discussed:
  - Doesn't need any derivatives! (sometimes...)
  - Can be used with latent variables!
- Further advantages:
  - Guaranteed that likelihood will increase with each iteration
  - E or M step often very easy to implement
  - Closed form solutions to M step often exist

The problem with the EM algorithm…

1. The M-step may need numerical methods (often requiring derivatives!) if analytical solution to maximization doesn't exist
2. The E-step may also need a numerical method due to intractability!

If the M-step not tractable:

- We can perform one step of Newtons method [1]
  - Any strict local max. point of the observed likelihood locally attracts EM with this replacement step as it would regular EM and at the same rate of convergence
  - Close to the max. point it always produces an increase in the likelihood
  - With some modification it also exhibits global convergence properties similar to that of EM
- Could also perform GD or SGD
- Could also use a stochastic approximation algorithm [2][4] (Might not need derivatives!)
- Some examples of where the M-step are intractable are provided in [1]

If the E-step intractable:

- Stochastic approximation [2][3]
- Monte Carlo methods [4][5]
- The better solution depends on simulation cost vs. maximization cost

References

- [1]: Lange, K., 1995. A gradient algorithm locally equivalent to the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(2), pp.425-437.
- [2] Gu, M.G. and Li, S., 1998. A stochastic approximation algorithm for maximum-likelihood estimation with incomplete data. *Canadian Journal of Statistics*, *26*(4), pp.567-582.
- [3] Delyon, B., Lavielle, M. and Moulines, E., 1999. Convergence of a stochastic approximation version of the EM algorithm. *Annals of statistics*, pp.94-128.
- [4] Gu, M.G. and Zhu, H.T., 2001. Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), pp.339-355.
- [5] Wei, G.C. and Tanner, M.A., 1990. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, *85*(411), pp.699-704.