Yoel Ferdman

Prof Sterne

CS156 Fall 2017 Assignment 4: German Tanks and EM with Mixture of Gaussians

[M98, M508, M727, M520, K85, K58, K13, K7, K74, K75, K64, F225, F292, F241, F453, F464, F165, F182, F334, F88]

1. Write out the likelihood of observing a single tank.

The likelihood function of observing a single tank when we have many observations is derived from the total number of combinations of observing *k-1* observations all smaller than *m-1*, the largest value of our observed sample, denoted by *(m-1) choose (k-1)*. Then dividing this by the possible combinations of observing *k* observations from *n* total observations (n will be our flexible variable/parameter here: how many tanks there are from each factory) gives us our likelihood function denoted by:

$$\mathcal{L}(n) = [n \geq m] \frac{\binom{m-1}{k-1}}{\binom{n}{k}}$$

2. Derive the **maximum likelihood** formula for the total number of tanks given a dataset as above.

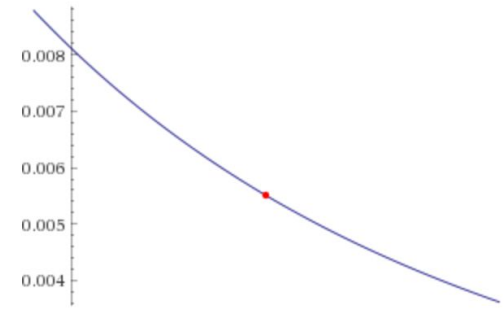
Each set of factories should be treated as independent tank problems, for obvious reasons. They might produce the same serial number from a different factory with a different letter identifying the factory.

For the M factory, we have *k=4* observations and *m=727* (highest serial number). This gives us the likelihood formula: L=(726 choose 3)/(n choose 4); where we are only interested in n values equal to and including 727.

Plugging this into wolfram alpha to find a maximum we see that the function only decreases at this point.

$$\max\left\{\frac{\binom{726}{3}}{\binom{x}{4}} \,\middle|\, x \geq 727\right\} = \frac{4}{727} \text{ at } x = 727$$

Plot:



(x from 647 to 807)

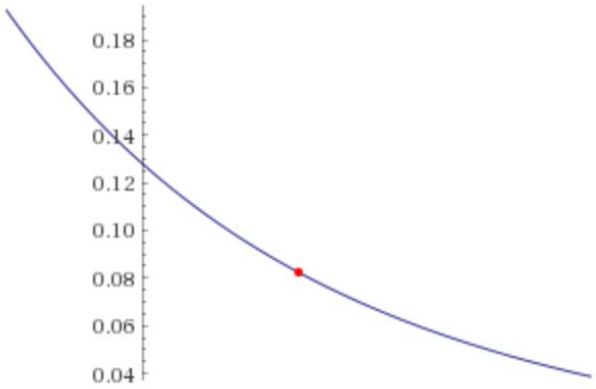
This makes intuitive sense that our maximum likelihood would be at our m=n value because when we see other problems like the train car problem where we only see one observation, we get a likelihood function 1/n which is constantly decreasing. Thinking about the chances that we would see all values lower than something that is not our upper (n) bound is less likely than seeing all values lower than just our upper (n) bound.

We see similar behavior in the plots for the other factories:

For the K factory, we have *k=7* observations and *m=85* (highest serial number). This gives us the likelihood formula: L=(84 choose 6)/(n choose 7); where we are only interested in n values equal to and including 85.

$$\max\left\{\frac{\binom{84}{6}}{\binom{x}{7}} \,\middle|\, 85 \leq x \leq \infty\right\} \approx 0.0823458 \text{ at } x \approx 85.001$$

Plot:



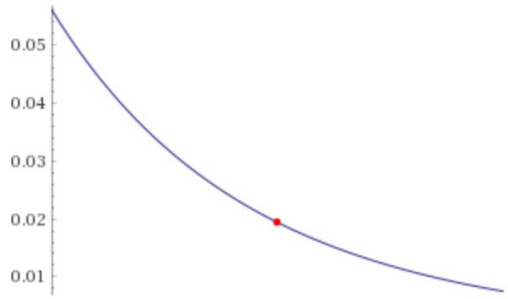
(x from 75.6 to 94.4)

And finally for the F factory we have 9 observations with a maximum serial, m value, at 464. Plugging this in we get: L=(463 choose 8)/(n choose 9); where we are only interested in n values equal to and including 464.

Wolfram gives the max value at n=464 of course.

$$\max\left\{\frac{\binom{463}{8}}{\binom{x}{9}} \,\middle|\, 464 \le x \le \infty\right\} = \frac{9}{464} \text{ at } x = 464$$

Plot:



(x from 413 to 515)

Use the formula that you derived in the previous question to estimate how many tanks there are given the dataset in the description.

M factory: 727

K factory: 85

F factory: 464

Total = 1276 tanks.

3. How strongly do you believe your estimate?

The statistical uncertainty, whose derivation is borrowed from Wikipedia below:

$$\begin{aligned}
\sigma^2 + \mu^2 - \mu &= \sum_n n(n-1) \cdot (N = n \mid M = m, K = k) \\
&= \sum_{n=m}^{\infty} n(n-1) \frac{m-1}{n} \frac{m-2}{n-1} \frac{k-1}{k-2} \frac{\binom{m-3}{k-3}}{\binom{n-2}{k-2}} \\
&= \frac{m-1}{1} \frac{m-2}{1} \frac{k-1}{k-2} \cdot \frac{\binom{m-3}{k-3}}{1} \sum_{n=m}^{\infty} \frac{1}{\binom{n-2}{k-2}} \\
&= \frac{m-1}{1} \frac{m-2}{1} \frac{k-1}{k-2} \frac{\binom{m-3}{k-3}}{1} \frac{k-2}{k-3} \frac{1}{\binom{m-3}{k-3}} \\
&= \frac{m-1}{1} \frac{m-2}{1} \frac{k-1}{k-3}
\end{aligned}$$

$$\begin{aligned}
\sigma &= \sqrt{\frac{m-1}{1} \frac{m-2}{1} \frac{k-1}{k-3} + \mu - \mu^2} \\
&= \sqrt{\frac{(k-1)(m-1)(m-k+1)}{(k-3)(k-2)^2}}
\end{aligned}$$

Gives us three different standard deviations for each factory:

M: m=727,k=4 σ=sqrt((3*726*724)/(4)) = 627.9

K: m=85, k=7 σ= sqrt ((6*84*79)/(4*25))=19.95

F: m=464, k=9 σ= sqrt((8*463*456)/(6*49)) = 75.8

The variance to mean ratio is: $\frac{\sigma^2}{\mu} = \frac{m-k+1}{(k-3)(k-2)}$ giving:

M: (727-4+1)/2 = 362

K: 79/20 = 3.95

F: 456/(6*7) = 10.9

Both the standard deviation and the variance to mean ratios indicate that these estimates are very uncertain. While given this data, we know that our results would give us the highest likelihood of accuracy for choosing our number of tanks, however it is not a strong belief that these are correct. We would either need more data or more assumptions to make this estimate more believable. If someone forced me to put money on this, I would go with the results we got, however I don't think I would win any money.

Mixture of Gaussians:

1. Using your generic equations (derived from the pre-class work on expectation maximization) show what the equations would be for a mixture of Gaussian distributions.

From http://www.ics.uci.edu/~smyth/courses/cs274/notes/EMnotes.pdf we see that finite mixture models with K components can be shown as:

$$p(\underline{x}|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(\underline{x}|z_k, \theta_k)$$

where:

- The $p_k(\underline{x}|z_k, \theta_k)$ are *mixture components*, $1 \leq k \leq K$. Each is a density or distribution defined over $p(\underline{x})$, with parameters θ_k.
- $z = (z_1, \ldots, z_K)$ is a vector of K binary indicator variables that are mutually exclusive and exhaustive (i.e., one and only one of the z_k's is equal to 1, and the others are 0). z is a K-ary random variable representing the identity of the mixture component that generated $\underline{x}$. It is convenient for mixture models to represent z as a vector of K indicator variables.
- The $\alpha_k = p(z_k)$ are the mixture weights, representing the probability that a randomly selected $\underline{x}$ was generated by component k, where $\sum_{k=1}^{K} \alpha_k = 1$.

It follows that

For $\underline{x} \in \mathcal{R}^d$ we can define a Gaussian mixture model by making each of the K components a Gaussian density with parameters $\underline{\mu}_k$ and Σ_k. Each component is a multivariate Gaussian density

$$p_k(\underline{x}|\theta_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_k)^t \Sigma_k^{-1}(\underline{x}-\underline{\mu}_k)}$$

with its own parameters $\theta_k = \{\underline{\mu}_k, \Sigma_k\}$.

$$\alpha_k^{new} = \frac{N_k}{N}, \quad 1 \leq k \leq K.$$

With mixture weights:

$$\underline{\mu}_k^{new} = \left(\frac{1}{N_k}\right) \sum_{i=1}^{N} w_{ik} \cdot \underline{x}_i \quad 1 \leq k \leq K.$$

$$\Sigma_k^{new} = \left(\frac{1}{N_k}\right) \sum_{i=1}^{N} w_{ik} \cdot (\underline{x}_i - \underline{\mu}_k^{new})(\underline{x}_i - \underline{\mu}_k^{new})^t \quad 1 \leq k \leq K.$$

"After we have computed all of the new parameters, the M-step is complete and we can now go back and recompute the membership weights in the E-step, then recompute the parameters again in the E-step, and continue updating the parameters in this manner. Each pair of E and M steps is considered to be one iteration."

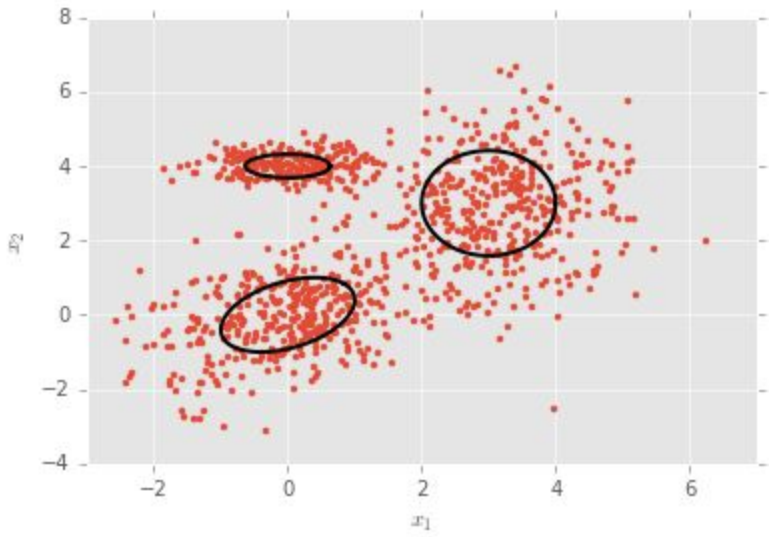
2. Find code online (or write your own) to generate a mixture of 2D Gaussians.
    a. http://www.nehalemlabs.net/prototype/blog/2014/04/03/quick-introduction-to-gaussian-mixture-models-with-python/
    b. http://yulearning.blogspot.kr/2014/11/einsteins-most-famous-equation-is-emc2.html
    c. https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html
3. Find code online (or write your own) that uses EM to cluster the data.

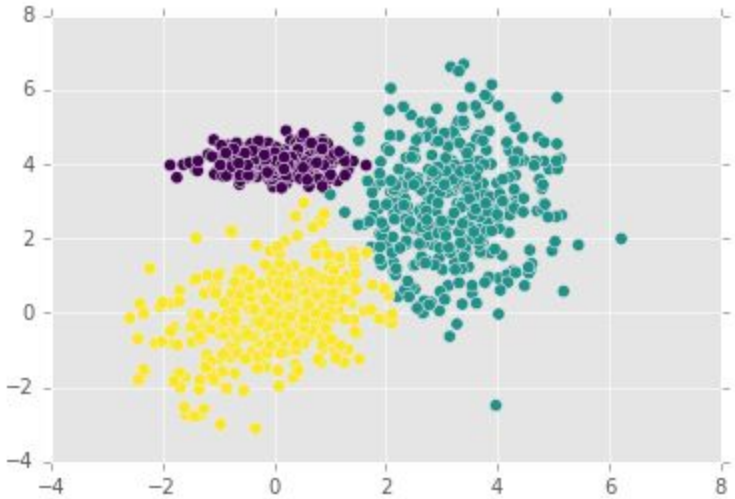
a. http://www.nehalemlabs.net/prototype/blog/2014/04/03/quick-introduction-to-gaussian-mixture-models-with-python/
b. http://yulearning.blogspot.kr/2014/11/einsteins-most-famous-equation-is-emc2.html
c. https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

4. Plot your generated data and the 2D-fitted Gaussians.

Using code adapted from Tomer E. and http://www.nehalemlabs.net/prototype/blog/2014/04/03/quick-introduction-to-gaussian-mixture-models-with-python/

https://gist.github.com/anonymous/f1e0b096719e23b93d373ae8669ab085

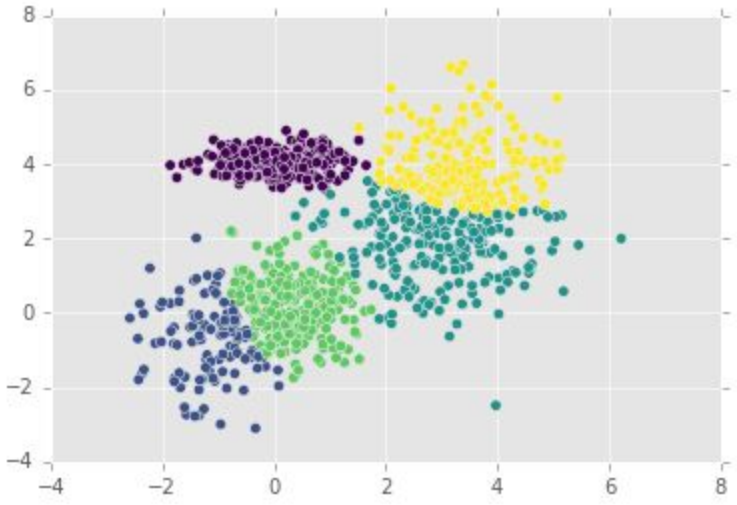




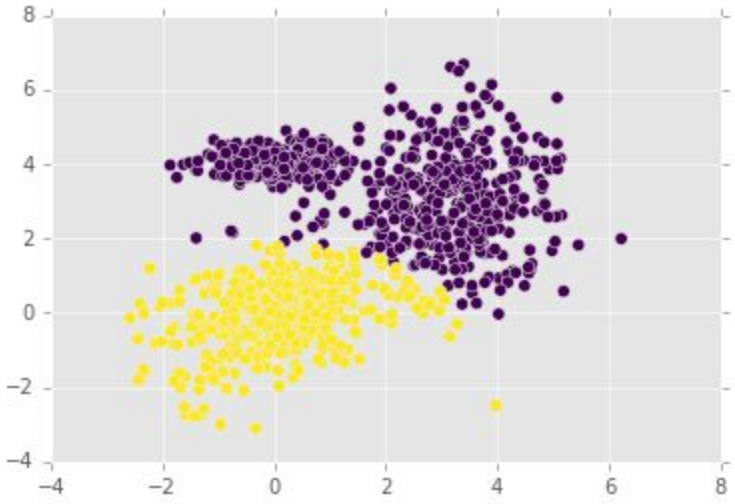
5. What happens if you try to fit a different number of clusters to the data? E.g. fit 5 clusters to data generated with only 2 clusters, or vice versa.

As we see in the graphs below, if we try to fit too many or too few clusters to the data then we will get the model's best incorrect guess. The core issue here is that when we don't know how many real clusters or groups there are in the data, then we have a hard time choosing the how many to pick unless that data is formatted really well from the start, which is rare. When we try to fit too many clusters, the model cuts up the data where it sees it most likely is vulnerable for a split (where it is least dense).

This plot uses too many clusters to fit the data.



This plot uses too few clusters.


8
6
4
2
0
−2
−4
−4
−2
0
2
4
6
8