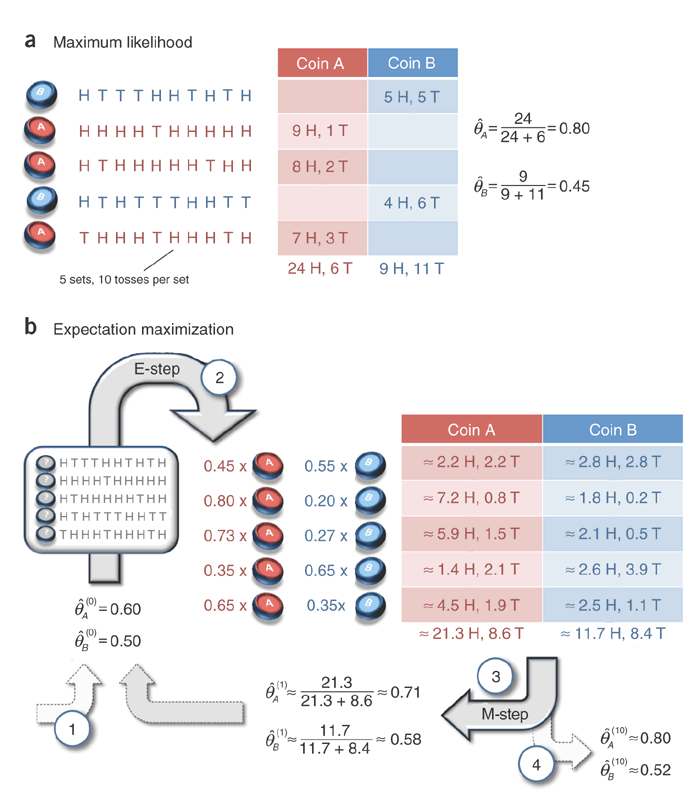
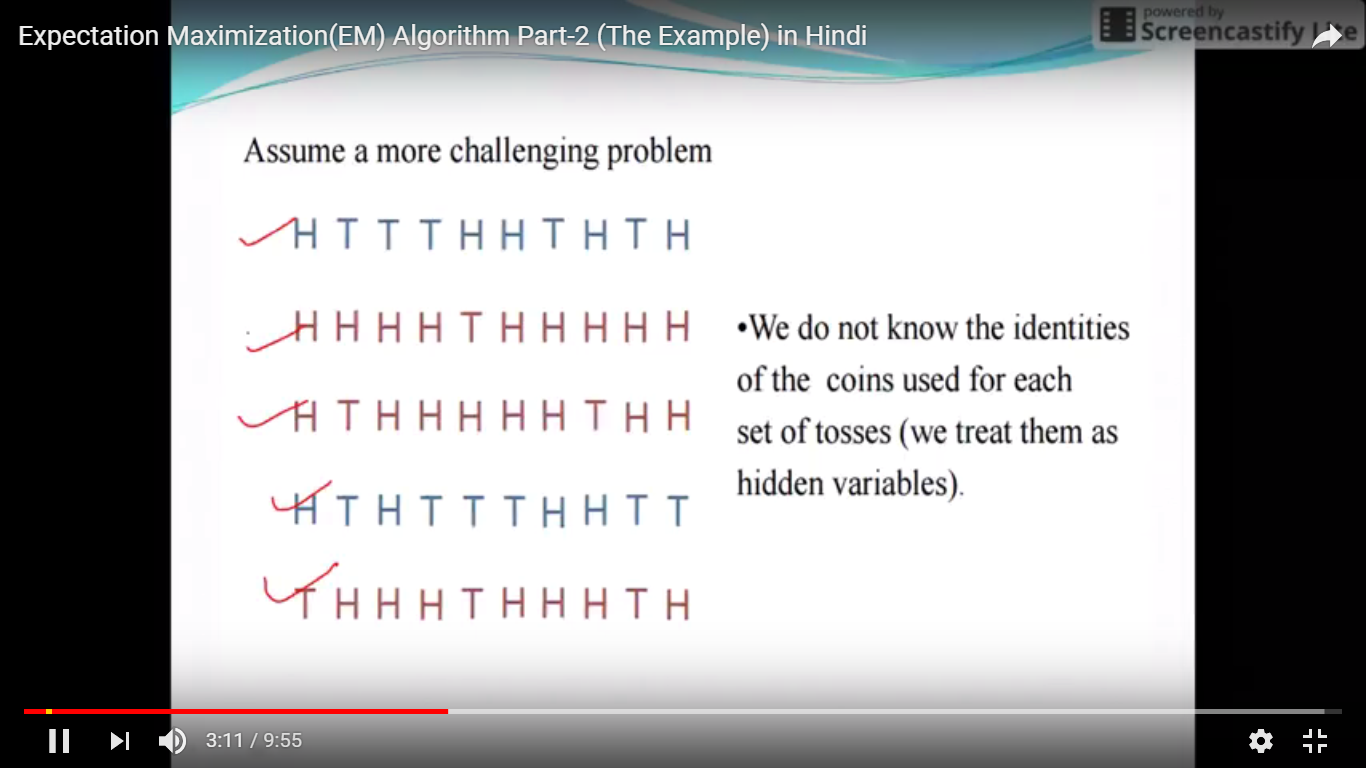
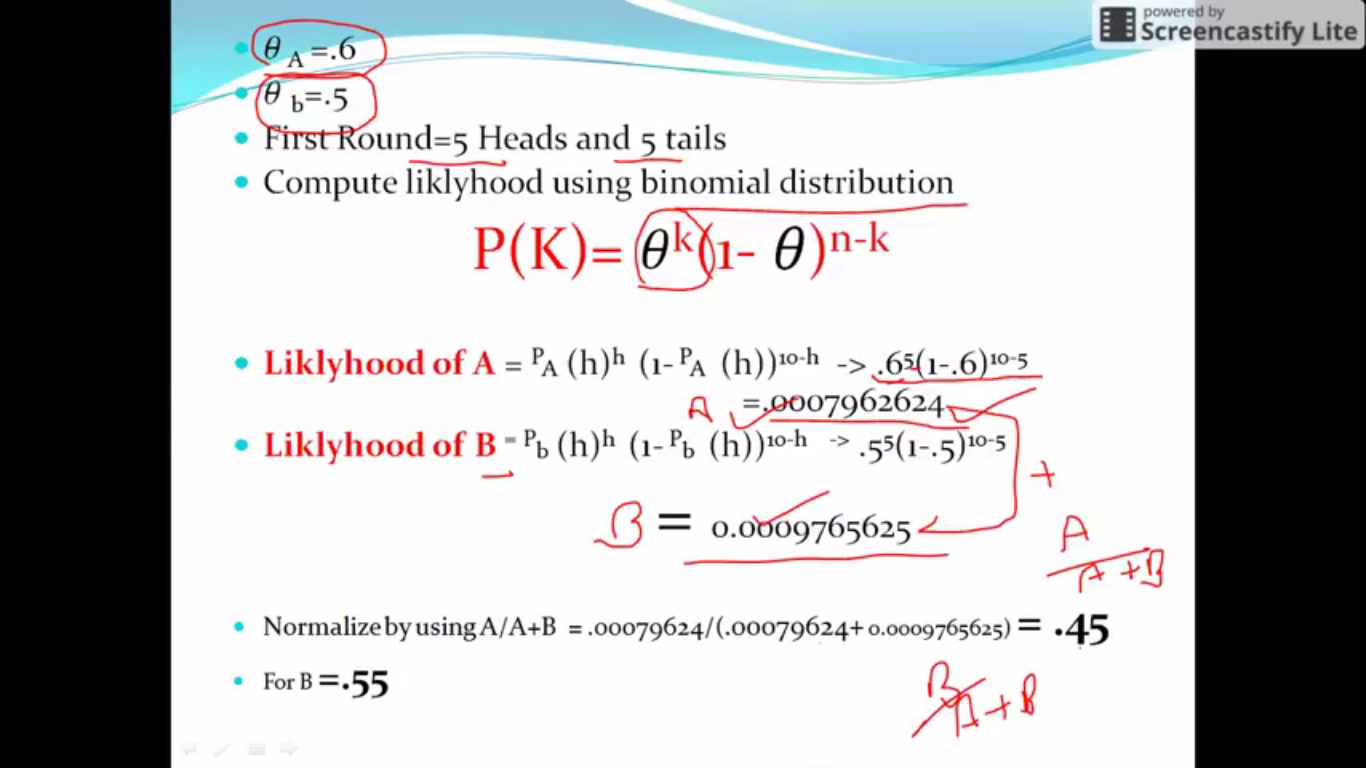
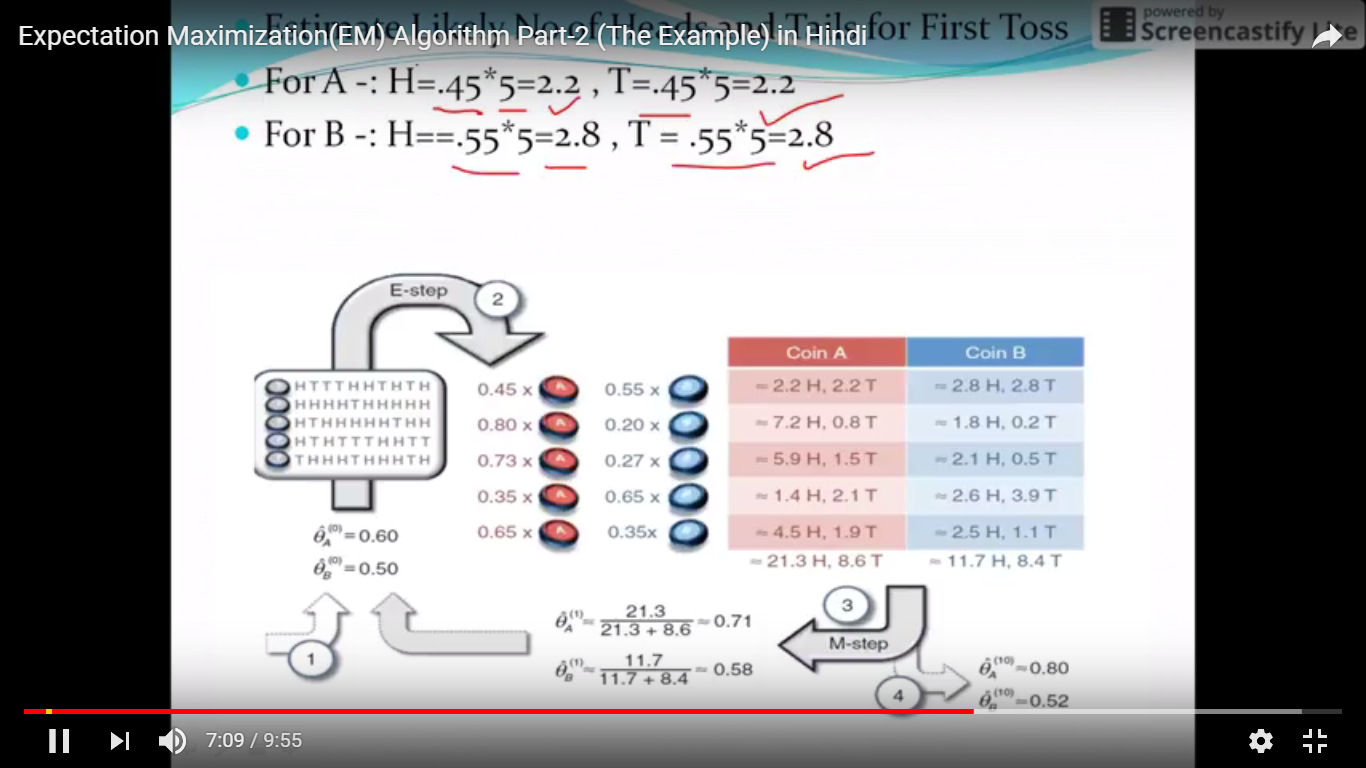
This is a recipe to learn EM with a practical and very intuitive 'Coin-Toss' example:

1. This is the schema where the coin toss example is explained:









1. You may have question marks in your head, especially regarding where the probabilities in the Expectation step come from.
2. Look at/run this code that I wrote in Python that simulates the solution to the coin-toss problem in the EM tutorial paper of item 1:
3. import numpy as np
4. import math
5. import matplotlib.pyplot as plt
6. ## E-M Coin Toss Example as given in the EM tutorial paper by Do and Batzoglou\* ##
7. def get\_binomial\_log\_likelihood(obs,probs):
8. """ Return the (log)likelihood of obs, given the probs"""
9. # Binomial Distribution Log PDF
10. # ln (pdf) = Binomial Coeff \* product of probabilities
11. # ln[f(x|n, p)] = comb(N,k) \* num\_heads\*ln(pH) + (N-num\_heads) \* ln(1-pH)
12. N = sum(obs);#number of trials
13. k = obs[0] # number of heads
14. binomial\_coeff = math.factorial(N) / (math.factorial(N-k) \* math.factorial(k))
15. prod\_probs = obs[0]\*math.log(probs[0]) + obs[1]\*math.log(1-probs[0])
16. log\_lik = binomial\_coeff + prod\_probs
17. return log\_lik
18. # 1st: Coin B, {HTTTHHTHTH}, 5H,5T
19. # 2nd: Coin A, {HHHHTHHHHH}, 9H,1T
20. # 3rd: Coin A, {HTHHHHHTHH}, 8H,2T
21. # 4th: Coin B, {HTHTTTHHTT}, 4H,6T
22. # 5th: Coin A, {THHHTHHHTH}, 7H,3T
23. # so, from MLE: pA(heads) = 0.80 and pB(heads)=0.45
24. # represent the experiments
25. head\_counts = np.array([5,9,8,4,7])
26. tail\_counts = 10-head\_counts
27. experiments = zip(head\_counts,tail\_counts)
28. # initialise the pA(heads) and pB(heads)
29. pA\_heads = np.zeros(100); pA\_heads[0] = 0.60
30. pB\_heads = np.zeros(100); pB\_heads[0] = 0.50
31. # E-M begins!
32. delta = 0.001
33. j = 0 # iteration counter
34. improvement = float('inf')
35. while (improvement>delta):
36. expectation\_A = np.zeros((len(experiments),2), dtype=float)
37. expectation\_B = np.zeros((len(experiments),2), dtype=float)
38. for i in range(0,len(experiments)):
39. e = experiments[i] # i'th experiment
40. # loglikelihood of e given coin A:
41. ll\_A = get\_mn\_log\_likelihood(e,np.array([pA\_heads[j],1-pA\_heads[j]]))
42. # loglikelihood of e given coin B
43. ll\_B = get\_mn\_log\_likelihood(e,np.array([pB\_heads[j],1-pB\_heads[j]]))
44. # corresponding weight of A proportional to likelihood of A
45. weightA = math.exp(ll\_A) / ( math.exp(ll\_A) + math.exp(ll\_B) )
46. # corresponding weight of B proportional to likelihood of B
47. weightB = math.exp(ll\_B) / ( math.exp(ll\_A) + math.exp(ll\_B) )
48. expectation\_A[i] = np.dot(weightA, e)
49. expectation\_B[i] = np.dot(weightB, e)
50. pA\_heads[j+1] = sum(expectation\_A)[0] / sum(sum(expectation\_A));
51. pB\_heads[j+1] = sum(expectation\_B)[0] / sum(sum(expectation\_B));
52. improvement = ( max( abs(np.array([pA\_heads[j+1],pB\_heads[j+1]]) -
53. np.array([pA\_heads[j],pB\_heads[j]]) )) )
54. j = j+1
55. plt.figure();
56. plt.plot(range(0,j),pA\_heads[0:j], 'r--')
57. plt.plot(range(0,j),pB\_heads[0:j])

plt.show()