Gene Itable
[redacted] Probabilistic Graphical Models

Report for Expectation Maximization and Gradient Ascent

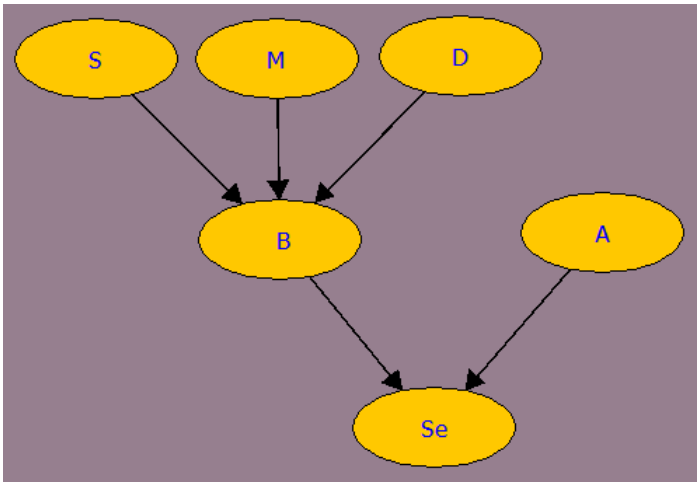
1 Data Set

1.1 Original data set

The mammographic mass data set used for my implementation of expectation maximization is retrieved from UC Irvine Machine Learning Repository[1]. The data consists of six attributes:

- BI-RADS assessment (from Shape, Margin and Density): 0 = incomplete, 1 = negative, 2 = benign, 3 = probably benign, 4 = suspicious, 5 = highly suggestive of malignancy, 6 = malignant
- Age: patient's age in years
- Shape (of the mass): 1 = round, 2 = oval, 3 = lobular, 4 = irregular
- Margin (of the mass): 1 = circumscribed, 2 = microlobulated, 3 = obscured, 4 = ill-defined, 5 = spiculated
- Density (of the mass): 1 = high, 2 = iso, 3 = low, 4 = irregular
- Severity (of the mass): 0 = benign, 1 = malignant

and will be denoted in symbols as B, A, S, M, D, Se respectively. An *instance* of a data is arranged as $[B, A, S, M, D, Se]$, e.g. $[5, 67, 3, 5, 3, 1]$. There are 961 instances, 825 are with completely specified attribute values, and 131 are with missing attribute values, e.g. $[3, 45, ?, 4, 3, 1]$. The dependencies (Bayesian network) of these attributes is shown in the figure below.



1.2 Simplified

The data set is further simplified by dividing the attributes B, A, S, M, D to only two values $\{0, 1\}$:

Table 1. Simplified data set

B	$\{1, 2, 3\} \to 0$, $\{4, 5, 6\} \to 1$
A	$\{\leq 40\} \to 0$, $\{> 40\} \to 1$
S	$\{1, 2\} \to 0$, $\{3, 4\} \to 1$
M	$\{1, 2\} \to 0$, $\{3, 4, 5\} \to 1$
D	$\{1, 2\} \to 0$, $\{3, 4\} \to 1$

[1] http://archive.ics.uci.edu/ml/datasets/mammographic+mass

There are five instances with $B = 0$ (incomplete) which we cannot simply group it with $B = \{1, 2, 3\}$ that are of a "benign" class. For simplicity, we remove these instances from the data set which now has 956 instances. These simplifications are only for easier implementation and discussion, and are not justified medically. Nonetheless, the results where S, M, D have their original dimensions will be presented later. The data set is divided into two, the **training data set** and the **test data set**. The former set is further divided into **complete training data set**, i.e. all attribute values are completely specified, and **incomplete training data set**, i.e. some attribute values are missing.

1.3 Calculating probabilities from the complete training data set

We calculate for the probabilities $P_A(a)$, $P_S(s)$, $P_M(m)$, $P_D(d)$ and conditional probabilities $P_B(b|s, m, d)$ and $P_{Se}(se|b, a)$ by simply counting their frequencies with respect to the complete training data set. They will be used as initial parameters;

- $\theta_{a_0} = P_A(a_0)$
- $\theta_{s_0} = P_S(s_0)$
- $\theta_{m_0} = P_M(m_0)$
- $\theta_{d_0} = P_D(d_0)$
- $\theta_{b_0|s_0,m_0,d_0} = P_B(b_0|s_0, m_0, d_0)$, ..., $\theta_{b_0|s_1,m_1,d_1} = P_B(b_0|s_1, m_1, d_1)$
- $\theta_{se_0|b_0,a_0} = P_{Se}(se_0|b_0, a_0)$, ..., $\theta_{se_0|b_1,a_1} = P_{Se}(se_0|b_1, a_1)$

for expectation maximization (EM) and gradient ascent (GA) algorithm.

2 Implementation of expectation maximization

2.1 The algorithm

Let us denote the initial parameters collectively as $\theta^0 = \{\theta^0_{a_0}, \theta^0_{s_0}, ..., \theta^0_{b_0|s_0,m_0,d_0}, ..., \theta^0_{se_0|b_1,a_1}, ...\}$, and the jth instance in the incomplete training data set as O_j, e.g. $O_{j'} = [1, 0, 0, ?, 1, 1]$. The EM algorithm goes as follows:
Expectation step (E-step) – For each data case O_j and each family X, U, compute the joint distribution $P(X|U; O_j, \theta^0)$. For our implementation, these would be $P_A(A; O_j, \theta^0)$, $P_S(S; O_j, \theta^0)$, $P_M(M; O_j, \theta^0)$, $P_D(D; O_j, \theta^0)$, $P_B(B|S, M, D; O_j, \theta^0)$, $P_{Se}(Se|B, A; O_j, \theta^0)$. We then compute for the *expected sufficient statistics* for each x, u as

$$M_{\theta^0}[X|U] = \sum_j P(X|U; O_j, \theta^0) \tag{1}$$

Example: We are to calculate $M_{\theta^0}[B|S, M, D]$ and we have the instance $O_j = [1, 1, 1, ?, 1, 0]$, there would be contributions of the form

$$\left(M_{\theta^0}[b_1|s_1, m_0, d_1]\right)_j = \theta^0_{se_0|b_1,a_1}\theta^0_{a_1}\theta^0_{b_1|s_1,m_0,d_1}\theta^0_{s_1}\theta^0_{m_0}\theta^0_{d_1}$$

$$\left(M_{\theta^0}[b_1|s_1, m_1, d_1]\right)_j = \theta_{se_0|b_1,a_1}\theta_{a_1}\theta_{b_1|s_1,m_1,d_1}\theta_{s_1}\theta_{m_1}\theta_{d_1} \tag{2}$$

Hence, we calculate for $M_{\theta^0}[A], M_{\theta^0}[S], M_{\theta^0}[M], M_{\theta^0}[D]$, which are 2×1 matrices, $M_{\theta^0}[B|S, M, D]$ which is a $2 \times (2 \cdot 2 \cdot 2)$ matrix and $M_{\theta^0}[Se|B, A]$ which is a $2 \times (2 \cdot 2)$ matrix.
Maximization step (M-step) – Here, we update the θ^0 parameters via

$$\theta^1_{x|u} = \frac{M_{\theta^0}[x|u]}{M_{\theta^0}[u]} = \frac{M_{\theta^0}[x|u]}{\sum_x M_{\theta^0}[x|u]} \tag{3}$$

in the implementation we make use of the second equality.

We repeat E-step then M-step until the θ parameters converges to a value.

2.2 Testing the parameters with the test data

Let us denote the parameters after convergence as θ^f. Our goal is if we can generate predictions with θ^f, particularly on B and Se given $A = a, S = a, M = m, D = d$. We treat $\{a, s, m, d\}$ as evidence so we set their parameters to $\theta^f_a = \theta^f_s = \theta^f_m = \theta^f_d = 1$. For the incomplete case where an attribute is missing, then we do not set its parameter to 1. We then determine the probability $P(b_0|S = s, M = m, D = d)$

$$P_B(b_0|s, m, d) = \sum_S \sum_M \sum_D \theta^f_{b_0|S,M,D} \theta^f_S \theta^f_M \theta^f_D \tag{4}$$

and the probability $P(Se|B, A = a)$

$$P_{Se}(se_0|B, a) = \sum_B \sum_A \theta^f_{se_0|b,a} \theta^f_b \theta^f_a \tag{5}$$

We then generate random numbers r_1 and r_2. If $r_1 < P_B(b_0|s, m, d)$ we predict that the BI-RADS assessment is $B = 0$, else $B = 1$. If $r_2 < P_{Se}(se_0|B, a)$ we predict that the Severity is $Se = 0$ (the mass is benign), else $Se = 1$ (the mass is malignant). We compare our prediction with the the value of B and Se in the instance of data and count how many times our prediction agrees with the data. We repeat this several times on the test data set.

2.3 Results for the simplified data set

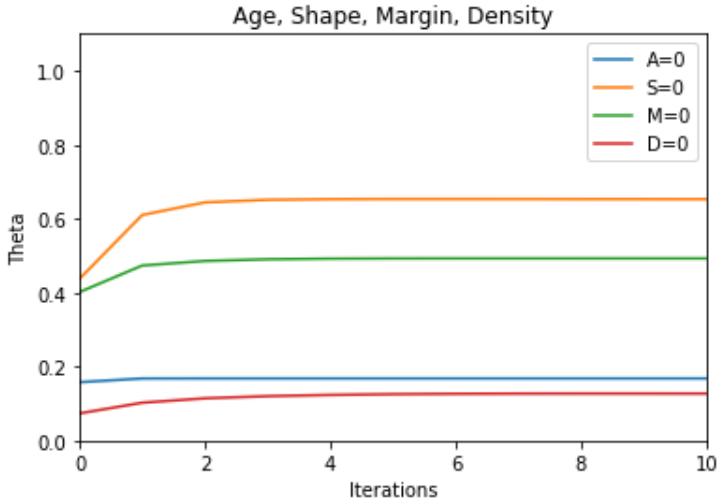
For this subsection, we discuss the results of our implementation with the simplified data set, see table 1, and see expectation_maximization.py for the code. Let us consider the case with 766 training data: 640 complete training data and 126 incomplete training data. From the complete training data, we determine our initial θ^0 parameters and perform the EM algorithm for 100 iterations.

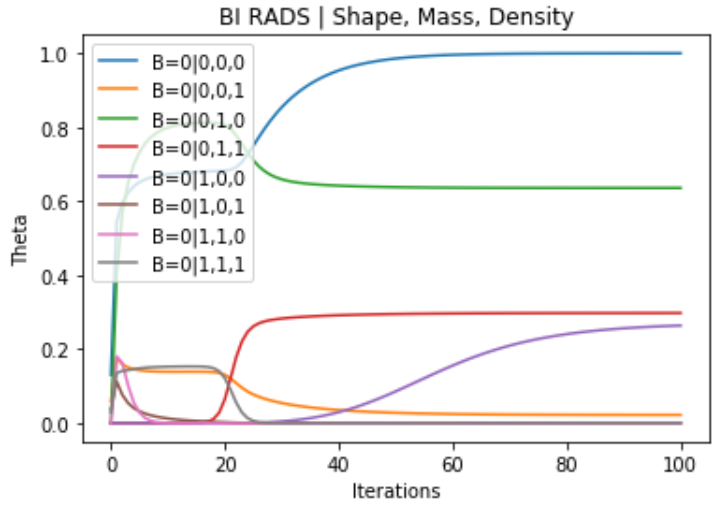
	θ^0	θ^f
$A = 0$	0.1578	0.1680
$S = 0$	0.4375	0.6842
$M = 0$	0.4016	0.4788
$D = 0$	0.0734	0.1232

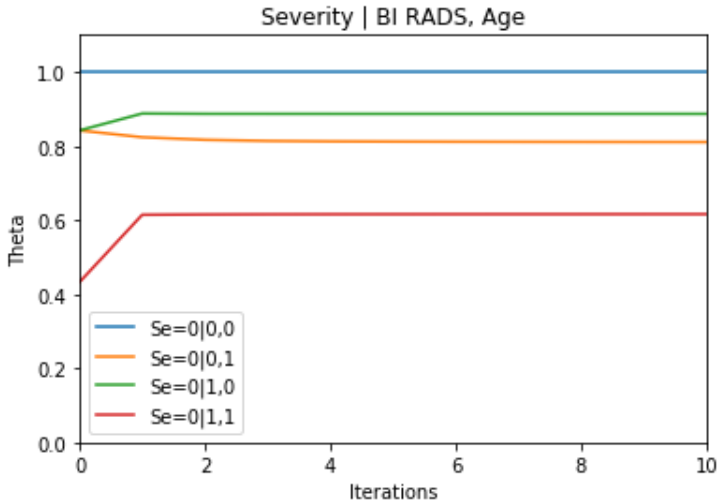
$B = 0\|S, M, D$	θ^0	θ^f
0,0,0	0.1304	0.9999
0,0,1	0.0594	0.0220
0,1,0	2×10^{-10}	0.6359
0,1,1	2×10^{-11}	0.2976
1,0,0	2×10^{-10}	0.2637
1,0,1	0.0370	0.
1,1,0	7×10^{-11}	0.
1,1,1	0.0287	0.

$Se = 0\|B, A$	θ^0	θ^f
0,0	1.0	1.0
0,1	0.8421	0.8032
1,0	0.8421	0.8868
1,1	0.4327	0.6178

The cases of A, S, M, D and $Se|B, A$ shows quicker convergence compared to $B|S, M, D$:







For testing with the test data set (190 instances), we conducted 500 repetitions of the prediction scheme in Sec. 2.4. We do this because of the random nature of our scheme. We obtain 88.6% agreement with the data on predicting B and 53.8% agreement with the data on predicting Se. Note that this 53.8% agreement should not discredit the model since it is stated that 70% of mass diagnosed by physicians to be malignant turned out to be benign upon biopsy (see mammomagraphic_masses_names.txt).

For different amount of instances in the training data set and test data set, the result is shown in the table below.

(Train, Test)	Agreement on B	Agreement on Se
(479, 477)	91.0%	53.7%
(638, 318)	90.6%	53.9%
(766, 190)	88.6%	53.8%
(861, 95)	90.7%	53.0%

2.4 Results for the original data set

Here, we do not have the simplicity to present our results in graphs the same way as in the simplified case because of the larger number of states per node. For example, the $P(B|S, M, D)$ is represented by a $6 \times (4 \cdot 5 \cdot 4)$ matrix. When θ_f is achieved, we test θ_f with respect to the test data. We change our prediction scheme for $B(= 1, 2, ..., 6)$ which now takes six values. Meanwhile, predicting Se is unchanged. For the code, see expectation_ maximization_ v2.py.

For B, we generate a random number r, then:

- if $r < P_B(b_0|s, m, d)$, then $B = 1$;
- if $P_B(b_0|s, m, d) \leq r < \sum_{b=b_0}^{b_1} P_B(b|s, m, d)$, then $B = 2$;
- if $\sum_{b=b_0}^{b_1} P_B(b|s, m, d) \leq r < \sum_{b=b_0}^{b_2} P_B(b|s, m, d)$, then $B = 3$; ...
- if $\sum_{b=b_0}^{b_4} P_B(b|s, m, d) \leq r < 1$, then $B = 6$.

We take 500 repetitions of this scheme over the test data for the results summarized below.

(Train, Test)	Agreement on B	Agreement on Se
(479, 477)	58.2%	47.0%
(638, 318)	58.3%	48.1%
(766, 190)	57.3%	47.4 %
(861, 95)	63.0%	48.4%

Compared to the simplified set, the agreements decrease. This may be attributed to the fact that we now have more states in B. If we predicted $B = b'$ with probability p, then we have $(1-p)$ probability that $B =$ not b' which is any of the possible five (not b') value.

3 Implementation of gradient ascent

3.1 The algorithm

The only difference of GA with EM is how it updates the parameters, which is the M-step in EM algorithm. For the first step of GA, we do the same E-step done in EM, i.e. calculating

$$M_{\theta^0}[X|U] = \sum_j P(X|U; O_j, \theta^0) \tag{6}$$

for each x, u. See the discussion of E-step in Section 2.1.

For the next step is to calculate the gradient of the log-likelihood, since we simplified the data set into binary values the gradient evaluate simply using chain rule, [2]

$$\frac{\partial l(\theta : \mathcal{D})}{\partial \theta_{a_0}} = \frac{M_{\theta_0}[A = a_0]}{\theta^0_{a_0}} - \frac{M_{\theta_0}[A = a_1]}{\theta^0_{a_1}} \tag{7}$$

same structure for S, M and D. If this quantity is positive, we increase $theta^0_{a_0}$, else we decrease $theta^0_{a_0}$, and we take the new value as $\theta^1_{a_0}$. Same goes for S, M and D, also for $B|S, M, D$ and $Se|B, A$

[2] this is line 13 and 14 in Algorithm 19.1 of Koller & Friedman.

only that we have the set

$$\begin{aligned}
\frac{\partial l(\theta:\mathcal{D})}{\partial \theta_{b_0|s_0,m_0,d_0}} &= \frac{M_{\theta_0}[B=b_0|S=s_0,M=m_0,D=d_0]}{\theta^0_{b_0|s_0,m_0,d_0}} - \frac{M_{\theta_0}[B=b_1|S=s_0,M=m_0,D=d_0]}{\theta^0_{b_1|s_0,m_0,d_0}}, \\
&\vdots \\
\frac{\partial l(\theta:\mathcal{D})}{\partial \theta_{b_0|s_1,m_1,d_1}} &= \frac{M_{\theta_0}[B=b_0|S=s_1,M=m_1,D=d_1]}{\theta^0_{b_0|s_1,m_1,d_1}} - \frac{M_{\theta_0}[B=b_1|S=s_1,M=m_1,D=d_1]}{\theta^0_{b_1|s_1,m_1,d_1}} \quad (8) \\
\frac{\partial l(\theta:\mathcal{D})}{\partial \theta_{se_0|b_0,a_0}} &= \frac{M_{\theta_0}[Se=se_0|B=b_0,A=a_0]}{\theta^0_{se_0|b_0,a_0}} - \frac{M_{\theta_0}[Se=se_1|B=b_0,A=a_0]}{\theta^0_{se_1|b_0,a_0}}, \\
&\vdots \\
\frac{\partial l(\theta:\mathcal{D})}{\partial \theta_{se_0|b_1,a_1}} &= \frac{M_{\theta_0}[Se=se_0|B=b_1,A=a_1]}{\theta^0_{se_0|b_0,a_0}} - \frac{M_{\theta_0}[Se=se_1|B=b_1,A=a_1]}{\theta^0_{se_1|b_1,a_1}}. \quad (9)
\end{aligned}$$

The way we decide how to increase or decrease the necessary parameter is in epsilon function in line 297-330 of gradient_ascent.py.

We repeat these steps until convergence and test them with the test data the same way as in EM, Section 2.2.

3.2 Results for the simplified data set

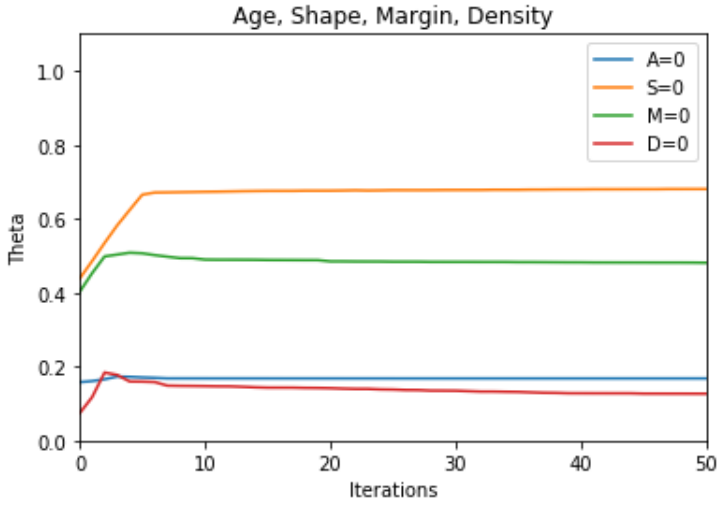
We discuss the results of our implementation with the simplified data set, see table 1, and see gradient_maximization.py for the code. We consider the same case, i.e. 766 training data: 640 complete training data and 126 incomplete training data. From the complete training data, we determine our initial θ^0 parameters and perform the GA algorithm for 700 iterations.

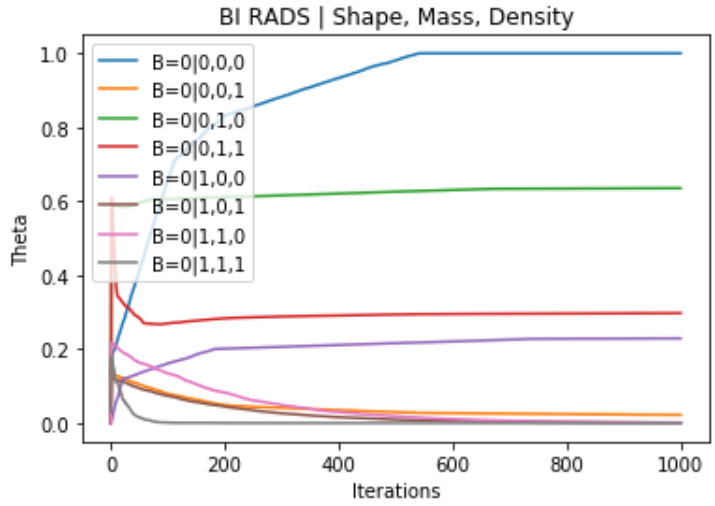
	θ^0	θ^f
$A=0$	0.1578	0.1680
$S=0$	0.4375	0.6844
$M=0$	0.4016	0.4790
$D=0$	0.0734	0.1233

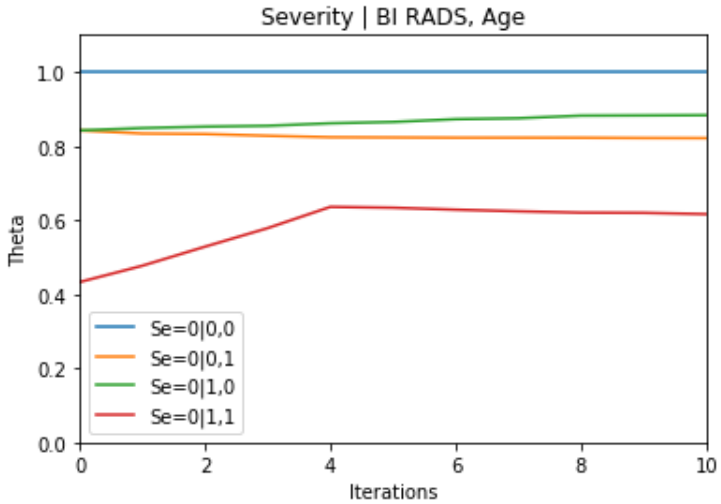
$B=0\|S,M,D$	θ^0	θ^f
0,0,0	0.1304	0.9999
0,0,1	0.0594	0.0222
0,1,0	2×10^{-10}	0.6330
0,1,1	2×10^{-11}	0.2975
1,0,0	2×10^{-10}	0.2007
1,0,1	0.0370	0.0008
1,1,0	7×10^{-11}	0.0039
1,1,1	0.0287	0.

$Se=0\|B,A$	θ^0	θ^f
0,0	1.0	1.0
0,1	0.8421	0.8037
1,0	0.8421	0.8868
1,1	0.4327	0.6178

The cases of A, S, M, D and $Se|B, A$ shows quicker convergence compared to $B|S, M, D$. The $B|S, M, D$ case also took longer iterations to converge in GA than in EM algorithm. We have agreements with the final values obtained with GA compared the results in EM, except significantly for $\theta^f_{b_0|s_1,m_0,d_0}$ which is 0.2007 in GA while it is 0.2637 in EM.







For different amount of instances in the training set and test set, we test the parameters we obtained the same way we did in EM. The results are summarized below.

(Train, Test)	Agreement on B	Agreement on Se
(479, 477)	90.5%	53.9%
(638, 318)	90.4%	53.8%
(766, 190)	88.6%	53.9%
(861, 95)	90.6%	53.0%

Unfortunately implementation on the original data set is a lot more complicated since we will no longer have the simplicity of the form of gradients (Eqs. (7-9)).