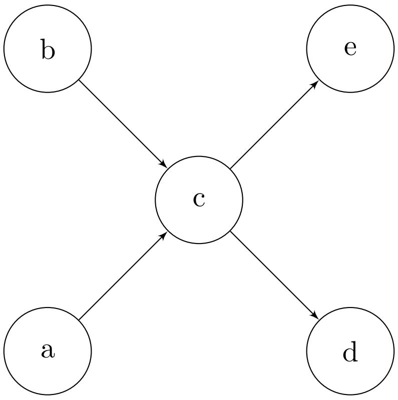
C S 391L: Machine Learning Assignment 5 **Julian Sia (EID: JAS7794)**
Prof. Dana Ballard Due Date: 09-May-16

Expectation Maximization (EM) in Directed Graphical Models

This report provides a brief description of experimental observations of expectation maximization used to recover a conditional probability distribution of a node in a directed graphical model.

Experimental Setup

To run our experiments, we consider the following graphical model structure below, running EM to recover the conditional probability distribution of node C. We believe this example is fairly robust example, as estimates of C rely on the the multiple children and parents it has a connection with. For simplicity, we assume that each of the variables have binary outcomes of 0 and 1 with 1.



Algorithm Description

For this algorithm, we use a form of standard belief propagation to do inference on our directed graphical model. Our algorithm, however, an be divided into discrete steps as relates to EM, expectation and maximization.

EXPECTATION (E)

In the expectation phase of our algorithm, we initiate our graph with a prior distribution over the graph. Namely, in this specific case we define priors for $P(A), P(B), P(C|AB), P(D|C), P(E|C)$. Since we wish to estimate the conditional distribution of A with a sample set and we assume that A data in the sample is missing, we can recover A by assigning a soft-label to the missing A. Namely, for each sample, (A, B, C, D, E) drawn from some sample set $S = \{(A_1, B_1, C_1, D_1, E_1), ..., (A_n, B_n, C_n, D_n, E_n)\}$ with n samples, where data is missing for C, we can compute $P(C = 1|A = k_1, B = k_2, D = k_3, E = k_4)$ and $P(C = 0|A = k_1, B = k_2, D = k_3, E = k_4)$ where the sample values are k_1,...,$k_4 \in \{0, 1\}$. We accomplish the above conditional probability Using these conditional probabilities, we can assign "soft" labels for each sample how likely A is 0 (i.e., $P(C = 0|A = k_1, B = k_2, D = k_3, E = k_4)$) rather than 1 (i.e., $P(C = 1|A = k_1, B = k_2, D = k_3, E = k_4)$).

MAXIMIZATION (M)

To recover the conditional probability of the estimated node (C in our case), we can extract the soft labels assigned to the estimated node (C in our example) and use the original data. Our graph structure allows us for more efficient estimation of conditional probabilities. Namely, given the nature of conditional probability, need only need to rely on the data of the parents of the estimated node and the soft labels assigned to the estimated node to compute the conditional probability distribution. This is approximated by frequency counts defined as the following for our case of estimating C:

$$P(C|A, B) = \frac{\#(C, A, B)}{\#(A, B)}$$

where # represents the frequency counts in the sample set where $(C = k_1) \cap (A = k_2) \cap (B = k_3)$ and $(A = k_2) \cap (B = k_3)$, where $k_1 ... k_3 \in \{0, 1\}$. In this case there are 4 possible combinations that A and B can take–namely, 00,01,10,and 11. More generally, we can do maximization for any N_{est} estimated node using the following formula:

$$P(N_{est}|Pa(N_{est})) = \frac{\#(N_{est}, Pa(N_{est}))}{\#(Pa(N_{est}))}$$

where $Pa(.)$ denotes the parent set of the node that it accepts as an argument

Combining E and M

Given a set of samples S and a node $\hat{N}$ we wish to estimate, we can use our previously defined Expectation and Maximization steps to find a stable conditional probability distribution for $\hat{N}$ using the following steps.

1. For each node in the graphical model, define a conditional distribution for each node.
2. Record the current conditional distribution of $\hat{N}$ as CPT_{old}
3. Run EXPECTATION with S for $\hat{N}$, producing output E_{output}.
4. Run MAXIMIZATION using E_{output} as input to produce a new conditional distribution for $\hat{N}$, denoted as CPT_{new}
5. Let $Pa(.)$ denote the parents of a given node in the graphical model. For each conditional probability estimate, $P(\hat{N}|Pa(\hat{N}))$ in CPT_{new} and $P(\hat{N}|Pa(\hat{N}))$ in CPT_{old}, compute the difference $\Delta = P(\hat{N}|Pa(\hat{N}))_{new} - P(\hat{N}|Pa(\hat{N}))_{old}$ for each configuration of parents (e.g., $P(C|A = 1, B = 0)_{new} - P(C|A = 1, B = 0)_{old}$).
6. If $\Delta < STOP$, then stop the algorithm and declare convergence. Otherwise, run steps 2-6 again until convergence.

As can be deduced from step 6, we use a hill climbing approach and stop when we approach a local maximum, as defined roughly by Δ.

Implementation

I implemented my EM algorithm for directed graphical models in Python 2.7. I wrote a variety of functions to run experiments testing my EM. I created the entire directed graphical model framework from scratch consisting of `directed_node.py` and `directed_graph.py` as the main elements. A testing harness `test_routines.py` was written to run the experiments described below. The user interacts with the graph API, instantiating a graph, then drawing edges, initializing the conditional probability tables, setting the conditional probability table priors, and then making a call to EM with a set of samples with a max termination bound specified for the run of EM.

Experiments

Purpose

In this set of experiments we are primarily motivated to empirically assess the behavior of EM in **runtime** (how many iterations it takes to converge given certain conditions) and the final state of the model at **termination time**–namely how conditional probability distributions converge (or diverge) under certain conditions discussed below. We take a strictly Bayesian interpretation of the probabilities–namely, we do not make explicit assumptions of a "true" distribution, but wish to see how choice of data updates our priors to give us a posterior estimate of our conditional probability distribution, or how effective our choice of priors affects how well our data gives us a posterior estimate of the conditional probability distribution.

Setup

We use the following the following sample set (10 samples) as a baseline (left) to do our computations and prior distribution (right three tables) over our previously discussed graph:

A	**B**	**C**	**D**	**E**
0	0	?	0	0
0	0	?	1	0
1	0	?	1	1
0	0	?	0	1
0	1	?	1	0
0	0	?	0	1
1	1	?	1	1
0	0	?	0	0
0	0	?	1	0
0	0	?	0	1

P(A)	**P(B)**
0.1	0.2

A	**B**	**P(C)**
1	1	0.9
1	0	0.6
0	1	0.3
0	0	0.2

C	**P(D)**	**P(E)**
1	0.9	0.8
0	0.2	0.1

We assume a $\Delta = 0.001$ stopping criteria out of convenience to limit the number of iterations that EM performs.

Experiment 1: Varying the Priors

In this experiment we vary the priors of our conditional distribution. For Experiment A we initialize our priors for C as the following: $P(C|A = 1, B = 1) = 0.9$, $P(C|A = 1, B = 0) = 0.6$,

$P(C|A=0,B=1)=0.3$, and $P(C|A=0,B=0)=0.2$ (the defined baseline parameters). For Experiment B, we initialize each probability in the prior distribution to a low value of 0.1 and in Experiment C, to a high value of 0.9. For this toy example with our fixed reference sample data which we defined above, EM took 15, 12, and 19 iterations to converge, respectively. As readily observable for the case of $A=0, B=1$ and $A=0, B=0$, we notice that although EM converges in a reasonable number of iterations, the sensitivity of the final agreed upon conditional probabilities differ widely depending on the initial starting conditions, which can result in a large variation in error given some true conditional probability distribution.

Table 1: Experiment A (Reasonable), Experiment B (Low), Experiment C (High)

A	**B**	**P(C)**
1	1	1.0
1	0	1.0
0	1	0.0016
0	0	0.3000

A	**B**	**P(C)**
1	1	1.0
1	0	1.0
0	1	0.0196
0	0	0.1000

A	**B**	**P(C)**
1	1	1.0
1	0	1.0
0	1	0.00217
0	0	0.9

Experiment 2: Varying Sample Data Quality

Experiments 2(a)(b): "Garbage" Data Experiments

In this set of experiments, we initialize a "garbage" sample in which all of the variables are 1 in the sample (Experiment A) and all the variables are 0 (Experiment B). As expected, this heavily biases the produced conditional probability distribution, resulting in distributions heavily biased towards 0 and 1 probability, as displayed below.

Table 2: Experiment A (All 0s), Experiment B (All 1s)

A	**B**	**P(C)**
1	1	0
1	0	0
0	1	0
0	0	0

A	**B**	**P(C)**
1	1	1
1	0	0
0	1	0
0	0	0

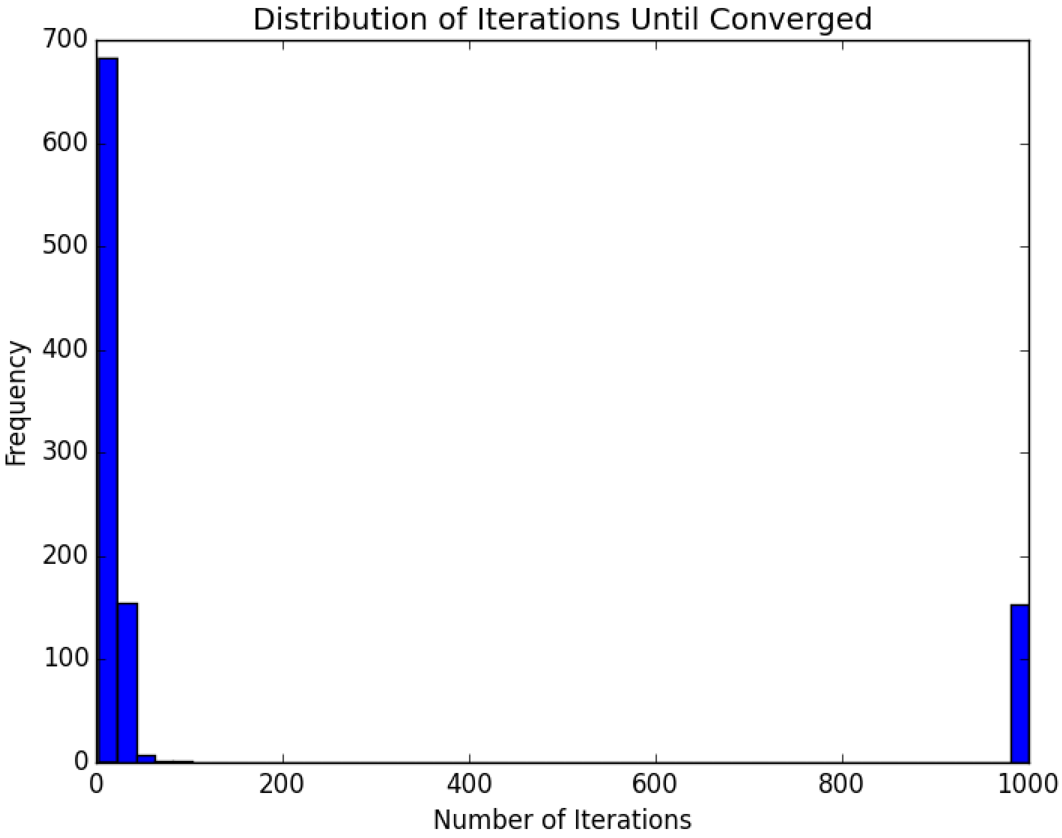
Experiment 2(c): Randomly Initialized Data Experiments

In addition to these "garbage" data experiments, we run 1000 simulations of EM with randomly initialized sample data sets of size 10 to see the effect that data quality has on convergence. On average, we note that EM converges in 166 iterations. However, we make the observation that EM generally converges within less than 100 iterations once we look at the run distribution. For some random initialization, it is interesting to note that convergence time took at least 1000 (if not greater) iterations. For runtime convenience, we set the maximum number of EM iterations at 1000, which may explain the fat tail of runs requiring 1000 iterations.

Experiment 3: Varying Sample Data Size

In this experiment, we increase the sample size ten-fold to 100 randomly initialized samples for our current graph and run 1000 simulations of EM In this case EM converges in an average of

Figure 1: Experiment 2(c) - Randomly Initialize Data Experiments



12 iterations. We notice that more data results in smoother distribution of outcomes (outlier runs disappear in this experiment). As our EM algorithm has more data to make predictions, it is sensible that convergence is achieved in fewer iterations.

Experiment 4: Leaf Node Recovery and Serial CPT Recovery

In this experiment we examine the effect of multiple approximations of CPT, namely recovering data for node D using estimated values of C. As our baseline, we use recovery of node D without estimating C. Using our toy sample, wee note that it takes 1 iteration to do recovery for D leaf node assuming we do not perform any estimation on C. However, with estimation of C (which requires 15 iterations of EM), there are 20 iterations required for EM to settle at an estimated value of 4. The increased number of iterations required for the multiple recovery case is perhaps an artifact of the soft-labeling scheme used to estimate the conditional probabilities of the C.

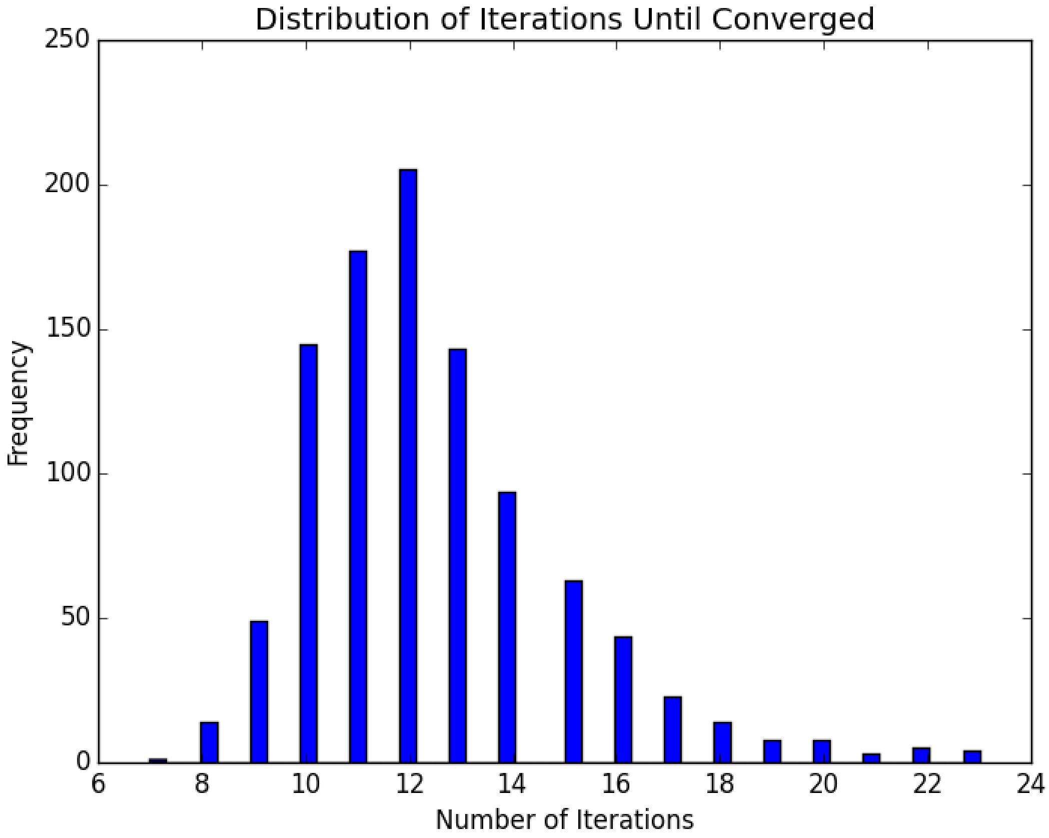
Table 3: Recovery of D (No Estimation), Estimated C, Recovery of D (C Estimation

C	**P(D)**
1	0.900
0	0.200

A	**B**	**P(C)**
1	1	0.9
1	0	0.6
0	1	0.3
0	0	0.2

C	**P(D)**
1	0.90
0	0.9972

Figure 2: Experiment 3 - Varying Sample Data Size



Future Work

Better Stopping Criteria

For our case of EM, we use a naive stopping criteria in which we stop if all of the conditional probabilities do not change by a large Δ–essentially, a naive hill climbing approach. It may be more useful to define Δ as the change in the likelihood function and adjust the stopping criteria accordingly.

Large Graph EM

Given that the explicit enumeration of the conditional probability distribution of each of the nodes grows at a factor of 2^n, where n is the number of parents that a node has in a directed graph (assuming strictly binary outcomes - for a more general case k^n, where k is the number of potential discrete outcomes that a node can take), inference becomes much more difficult. Very quickly, this approach may fail to scale in dense graphs where nodes have many parents. However, for sparse graphs with a small number of potential outcomes, the inference becomes more tractable.

Attachments

`sia_hw5_code.zip`

References

[1] Bishop, Christopher J. 2006. *Pattern Recognition and Machine Learning.* Springer Science+Business Media, LLC, 2006 [cited 15 February 2016].

[2] *Learning Bayesian Networks.* University of Wicsconsin. `http://pages.cs.wisc.edu/ dpage/cs760/BNall.pdf`[cited 8 May 2016].