Final Project: Crowd-sourcing

IE 598 Inference in Graphical Models

Kiran Koshy . Thekumparampil[1] Pavankumar Reddy. Muddireddy[1]

[1]Department of Electrical and Computer Engineering
University of Illinois

Inference in Graphical Models
Final Project Presentation, 2015

Outline

1. Model
   - Graphical Model for crowdsourcing
   - Model for Expectation Maximization
2. Belief Propagation update Equations
   - Message passing algorithm
3. Expectation Maximization
   - Objective
   - Design of the task Assignment
   - Results with EM
   - Observations from plain EM

Outline

1. Model
   - Graphical Model for crowdsourcing
   - Model for Expectation Maximization
2. Belief Propagation update Equations
   - Message passing algorithm
3. Expectation Maximization
   - Objective
   - Design of the task Assignment
   - Results with EM
   - Observations from plain EM

Likelihood of tasks labels and reliability parameters

- The joint likelihood for tasks labels(i) denoted by t_i and reliability parameters for worker(j) denoted by p_j given responses A_{ij} (to task i by worker j) is denoted by $\mu(t, p|A)$
- $$\mu(t, p|A) \propto \mu(t, p, A) \propto \mu(A|t, p)\mu(t)\mu(p)$$
  $$\mu(t, p|A) = \frac{1}{Z} \prod_{i \in [n]} F'(t_i) \prod_{j \in [m]} F(p_j) \prod_{(i,j) \in E} (\psi_{ij}(A_{ij}, p_j, t_j))$$
  where, $\psi_{ij}(A_{ij}, p_j, t_j) = \mathbb{I}(t_i = A_{ij})p_j + \mathbb{I}(t_i = -A_{ij})(1 - p_j)$
- Here $F'(t_i)$ is the bernoulli (0.75) prior of task labels and $F(p_j)$ is the beta distribution with $\alpha = 6$ and $\beta = 2$

Outline

Likelihood of tasks labels and reliability parameters

- The likelihood for tasks labels given responses A_{ij} using reliability parameters for workers as parameters for the EM is denoted by $\mu_p(t|A)$
- $$\mu_p(t|A) = \frac{1}{Z} \prod_{i \in [n]} F'(t_i) \prod_{(i,j) \in E} (\psi_{p_j}(A_{ij}, t_j))$$

  where, $\psi_{p_j}(A_{ij}, t_j) = \mathbb{I}(t_i = A_{ij}) p_j + \mathbb{I}(t_i = -A_{ij})(1 - p_j)$

Outline

1. Model
   - Graphical Model for crowdsourcing
   - Model for Expectation Maximization
2. Belief Propagation update Equations
   - Message passing algorithm
3. Expectation Maximization
   - Objective
   - Design of the task Assignment
   - Results with EM
   - Observations from plain EM

Quantization of beta distribution

- Since, the beta distribution is continuous and hence, the message passing would involve passing a continuous distribution from each worker to task and also integration at each task message to worker, the distribution is quantized to discrete points.
- In effect, this would be like having probability of each p_k belonging to an interval and the sum would be equivalent to approximate numerical integration.

Message passing algorithm

- From the joint distribution earlier the messages from different kinds of nodes in the graph are,
- Messages from worker to task,

$$\nu_{j\to i} \propto \prod_{k\in\partial j\backslash i} \sum_{t_k} F'(t_k)\psi_{kj}(t_k, p_j)\nu_{k\to j}(t_k)$$

- Messages from task to workers,

$$\nu_{i\to j} \propto \prod_{k\in\partial i\backslash j} \sum_{p_k} F(p_k)\psi_{kj}(t_i, p_k)\nu_{k\to i}(p_k)$$

Outline

Objective in EM

- The EM is formulated with p_j's as parameters, again to handle the continuous nature of the distribution.
- Hence, the objective would to find the parameters that would maximize the likelihood,
- $$p* = \underset{p}{\operatorname{argmax}}\, \mu_p(A) = \underset{p}{\operatorname{argmax}} \sum_{t \in \{-1,+1\}^n} \mu_p(t, A)$$

  where $\mu_p(t, A) = \frac{1}{Z} \prod_{i \in [n]} F'(t_i) \prod_{(i,j) \in E} (\psi_{p_j}(A_{ij}, t_j))$

Message passing algorithm

- $q_i(.)$ initialized to fraction of positive and negative responses respectively for i (majority voting).

$$p_j = \frac{\sum_{i \in \partial j} q(A_{ij})}{|\partial j|}$$

- Update equations:

$$q_i(t_i) \leftarrow \frac{F'(t_i) \prod_{j \in \partial i} \mu_{p_j}(A_{ij}|t_i)}{\sum_{t_i' \in -1,+1} F'(t_i') \prod_{j \in \partial i} \mu_{p_j}(A_{ij}|t_i')}$$

$$p_j \leftarrow \frac{\sum_{i \in \partial j} q(A_{ij})}{|\partial j|}$$

Outline

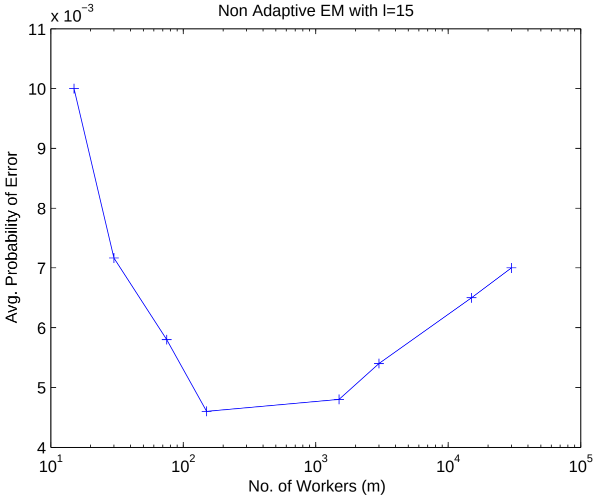
Design of the task Assignment

- Given a number of tasks and workers with l and r parameters, Erdős-Rényi is not used, rather a random (l, r) graph is generated by taking nl half edges and mr half edges and connecting a random permutation of them. This ensures that the degree distribution is uniform across both tasks and workers instead of bernoulli as in Erdős-Rényi graph.

Outline

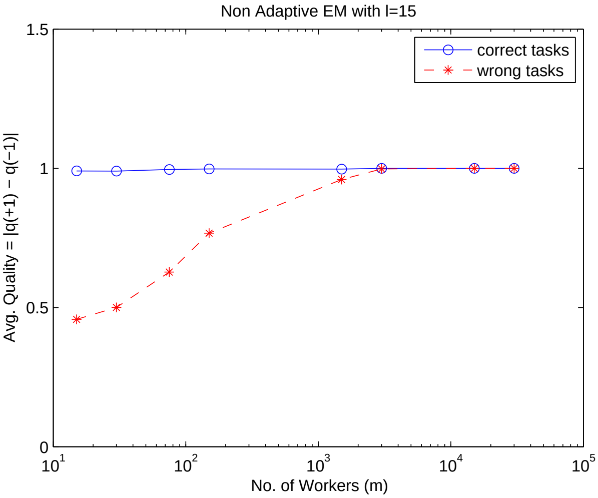
Results with plain EM

- The plot of error with number of workers,



Results with plain EM

- The plot of quality with number of workers,



Outline

Observations with plain EM

- The quality of a task estimate is defined as $|q(+1) - q(-1)|$, which indicates the amount of uncertainty in the prediction of a task label. So, low quality indicates, very high uncertainty.
- From the plot of quality distribution for misclassified tasks, it can be seen that the quality is a very good metric for classification error for a given task.
- So, any adaptation done could use quality of task estimates to assign more workers for a given tasks.

Observations with plain EM

- Additionally, one could define similar quality for the workers with $|2p_j - 1|$, which indicates how uncertain the worker was in giving the response for a given task.
- One could remove low quality workers. But its been observed that losing budget, even low quality, was very detrimental to the estimates. So, the task estimates were very sensitive to the their fanout.

Adaptive EM Algorithm

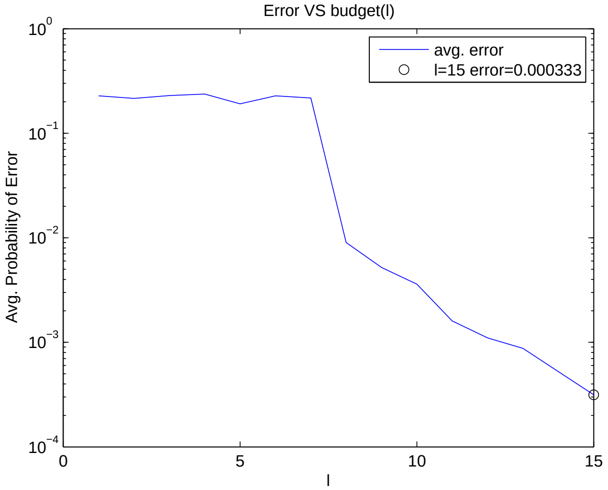
- Our adaptive algorithm progressively adds half of budget left at each iteration.
- After the each run of EM we take the bottom half of the tasks ranked by quality plus bottom hundredth of the tasks ranked by degree and add half of the budget left to these tasks by connecting them to new workers.
- For initialization estimates of tasks and workers are copied from the previous iterations
- This is done till budget is exhausted.

Observations in adapted EM

- As seen in the earlier plain EM, keeping the number of workers in the order of number of tasks was non-ideal. This was again observed in adapted EM.
- Could be because p_j estimate needs a large r which intern implies a small m for a given budget $n \times l$.
- Ideal estimates when $m = l$ or $m = 2l$.
- The adapted EM with more workers just for low quality workers was not giving some outliers with good quality but an incorrect estimate of label. Its been seen that degree of such nodes was low. (We start out with uniform degree but it changes as more wokers are added to some of the tasks). So, low degree tasks are also provided with more workers.

Final Results in adapted EM

- The plot of error with l,



Final Results in adapted EM

- The plot of error with l, The final error is 3.33×10^{-4} for $m = l$ and 2×10^{-4} for $m = 2l$