IE 598 Inference in Graphical Models

Final Project

The final consists in designing inference algorithms for a crowdsourcing problem given noisy responses from possibly unreliable workers, under the Dawid-Skense model. The objective is to minimize the average error probability

$$Pe(\hat{t}) \quad = \quad \frac{1}{n} \sum_{i=1}^{n} \mathbb{P}(t_i \neq \hat{t}_i) \tag{1}$$

given a fixed budget constraint on how many task-worker pairs can be assigned to get responses on.

Constraints

- there are n tasks with (hidden) binary true labels, i.e. $t = [t_1, \ldots, t_n] \in \{-1, +1\}^n$
- there are m workers parametrized by (hidden) random variables $p = [p_1, \ldots, p_m] \in [0, 1]^m$
- the responses are binary as per Dawid-Skene model such that if task t_i is assigned to worker p_j, then the outcome is

$$\mathbb{P}(A_{ij}|t_i, p_j) \quad = \quad \begin{cases} p_j & \text{if } A_{ij} = t_i \\ 1 - p_j & \text{if } A_{ij} = -t_i \end{cases} \tag{2}$$

- the true labels t_i's are generated i.i.d. from

$$t_i = \begin{cases} +1 & \text{with probabitliy } 3/4 \\ -1 & \text{with probability } 1/4 \end{cases} \tag{3}$$

- **Prior on p_j's.** each worker has reliability parameter p_j drawn i.i.d. according to the following distribution:

$$p_j \quad = \quad 0.1 + 0.9\, Z_j \ , \tag{4}$$

  where Z_j's are i.i.d. random variables drawn from the beta distribution with parameters α and β, i.e. $Z_j \sim \text{Beta}(\alpha, \beta)$ for $\alpha = 6$ and $\beta = 2$. In MATLAB, `p=0.1+0.9*betarnd(alpha,beta,m,1)` generates m i.i.d. random variables of this distribution. The beta distribution $\text{Beta}(\alpha, \beta)$ has pdf

$$f(z_j) \quad = \quad \frac{1}{B(\alpha, \beta)} z_j^{\alpha-1} (1 - z_j)^{\beta-1} \ ,$$

  for $z_j \in [0, 1]$, where $B(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$ is the normalization constant.

- in this project, we will use $n = 2,000$ tasks and you are free to use as many workers as you need, as long as each worker's parameter p_j's are i.i.d. drawn from the above distribution.
- **Budget constraint.** In this project you are free to choose how you assign edges, which is equivalent as deciding which task to assign to which worker, as long as the following conditions are met.

- the total number of edges must not exceed ℓn for $n = 2,000$ and for the values of $\ell \in \{1, 2, \ldots, 15\}$
- the task assignment should not use the ground truth on the (hidden) worker reliabilities p
- the task assignments can be done adaptively, by choosing the next assignment go a subset of the edges adaptively based on the edges placed and responses collected thus far
- one crucial rule is that a worker cannot be reused. This means that if a worker is assigned a certain number of tasks, and responded, you cannot assign more tasks to that worker. Instead, adaptation should be on the tasks, such that you can collect some responses on a task and then later adaptively choose to assign more (fresh) workers to that task.

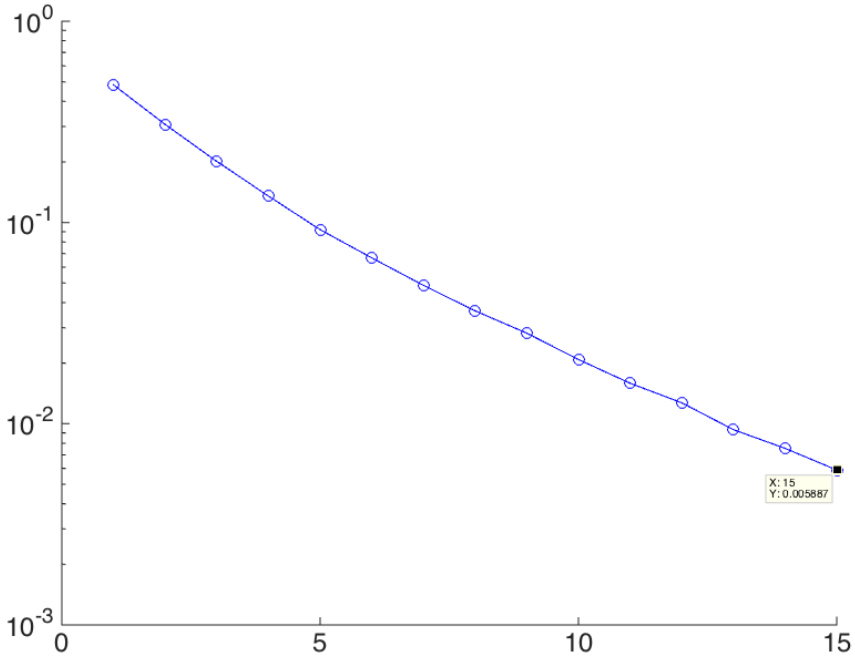
The objective is to design a possibly adaptive task assignment scheme, and corresponding inference algorithm for inferring t_i's, in order to minimize the average error probability Eq. (1). The goal is to design the task assignment and the inference algorithm that achieve error probability of 0.001 for $\ell = 15$.

Presentation of the results.

This is a team project for a team of two people (or less). You are required to write a short report on your design. The report will be posted on course website. Reports must have the following form:

1. A description of the design(s) of the task assignment.
2. A description of the inference algorithm(s).
3. A complete description of the program used for simulations (as a file).
4. A report of simulation results
   - plot the average probability of error (averaged over at least 100 instances per point) versus the budget as defined by ℓ for the values of $\ell \in \{1, 2, \ldots, 15\}$ (in a semi log plot using `semilogy()`).
   - convincingly show that the target error probability is achieved for $\ell = 15$. One way to do it is by showing that the mean + standard deviation is less than the target 0.001. One can compute this quantity by running the simulations N times, and let $P_k = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(t_i \neq \hat{t}_i)$ be the average error rate in k-th trial. Let P be the N dimensional vector storing P_k's. Then, in MATLAB "`mean(P) + (1/sqrt(N))*std(P)`" computes the mean + standard deviation for the estimated probability of error. You want to choose N large enough so that the second term is small. The goal is to design algorithms such that this quantity os smaller than 0.001 for $\ell = 15$.

For example, if we assign tasks according to Erdös-Renyi graph where each edge is sampled with probability ℓ/m for $m = 250$ workers, and apply the simplified BP algorithm (not using the information about the prior), the "mean + standard deviation" is plotted below, with error 0.005887 at $\ell = 15$.



Reports are due May 4th midnight, and each team will give a short presentation (10 minutes each) of their ideas and results in front of the class on Tuesday May 5th 12:30-2:00 PM.