- Crowd is large, anonymous, transient
  - Impossible to build up a trust relationship
  - Difficult to condition payment on correct answers (no groundtruth, any delay is annoying workers, …)
  - Need to exploit redundancy:
    - One tasks assigned to multiple workers
    - One worker realising more than one task
- This presentation: some (classical) ML approach to Learning with Crowds
  - Classical, not alicia(l)!

Models

- First Level: estimate the ground truth $y\downarrow i\uparrow$ from a set of { $y\downarrow i\uparrow k$ }
  - Ex: Majority Voting
- Second Level: models the worker's expertise
  - Simple "accuracy" parameter
  - "Confusion-matrix" set of parameters
  - Intra/Inter-consistency parameters
- Third Level: also models task difficulty
  - One parameter for worker's expertise and one parameter for task difficulty
- Of course, all parameters are unknown and should be identified together with:
  - Either the correct labels ($y\downarrow i\uparrow$)
  - Classification models to predict the correct labels for new crowdsourced instances

Level 1 : Estimating the correct label $y\downarrow i\uparrow$

- Assume *m* tasks, *n* workers, 1 task to *w* workers, 1 workers to *t* tasks (→*m.w=n.t*)
  - One-shot setting! (anonymous, transient workers)
- Basic inputs: $A\downarrow i,j$ (binary labels $y\downarrow i\uparrow : -1\ or +1$)
  - $A\downarrow i,j = 0$ if worker *j* didn't label task *i*
  - $A\downarrow i,j = y\downarrow i\uparrow$ with probability $p\downarrow j$
  - $A\downarrow i,j = -y\downarrow i\uparrow$ with probability (1-$p\downarrow j$)
  - $p\downarrow j$ not modelled but assumed i.i.d. from some unknown distributions with some properties (for convergence)
- Bipartite graph – HITS algorithm
  - Starts with $v\downarrow j\uparrow$=1/*w*
  - Iterate until convergence
    - $u\downarrow i = \sum j\uparrow \blacksquare A\downarrow i,j \ . \ v\downarrow j\uparrow$
    - $v\downarrow j = \sum i\uparrow \blacksquare A\downarrow i,j \ . \ u\downarrow i\uparrow$
  - Output: $y\downarrow i = sign(u\downarrow i)$

Level 1 : HITS Algo

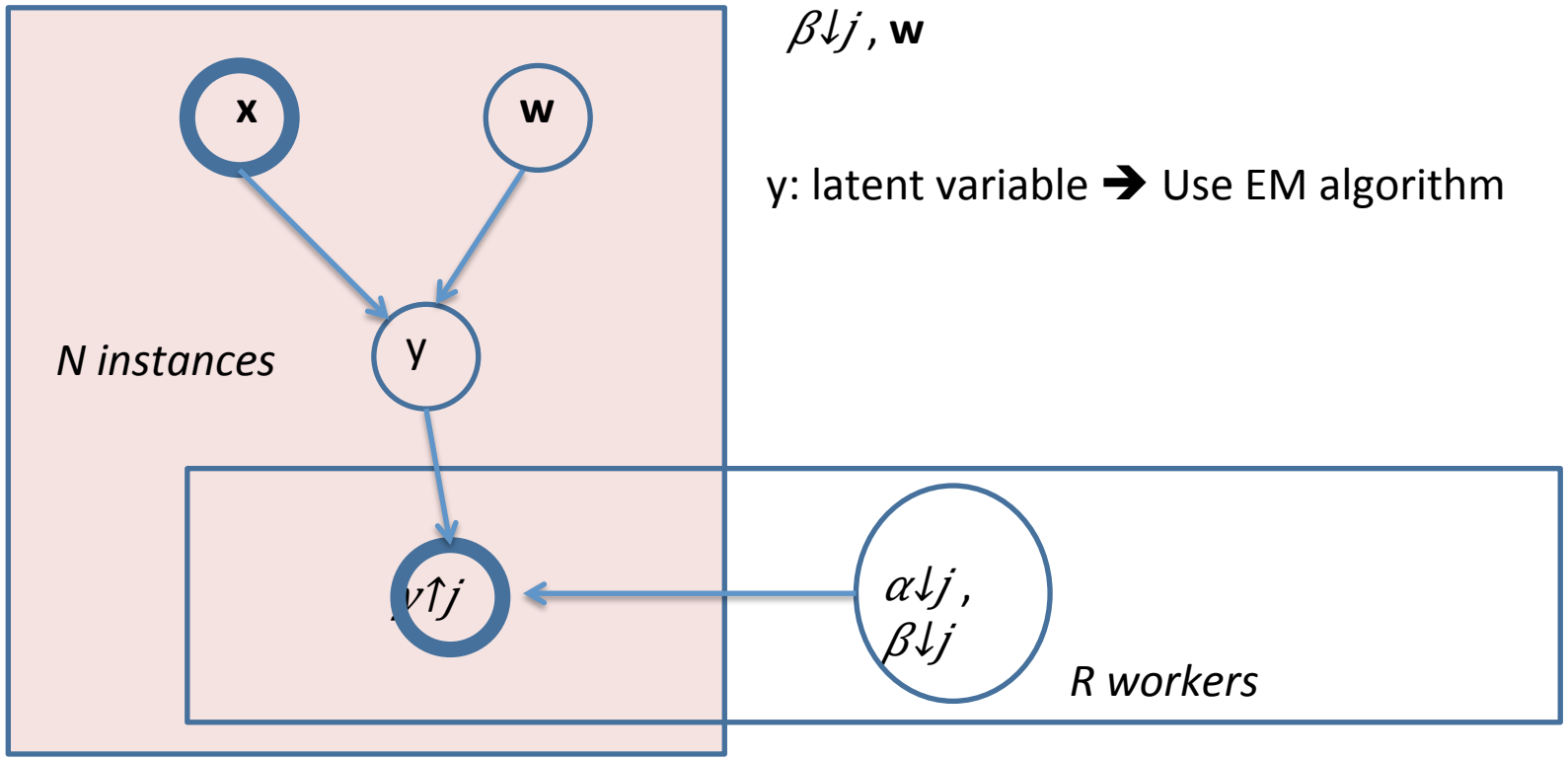
- Let z=E[(2 $p\downarrow j$ -1)] and q=E[(2 $p\downarrow j$ -1)^2]
  - q=0 → all random workers
  - q=1 → all workers are diligent
- Then, if z>0 and q^2(w-1).(t-1)>1

  P($y\downarrow i \neq y\downarrow i$)$\leq e\uparrow -wq/k$

- NB. Converges towards the leading left and right singular vectors of $A\downarrow i,j$
- The bounds can be translated in the *w* requirements to achieve a given accuracy, given the average quality *q*.

Level II : Individual Worker Model

- $p y^{(j)} y = (\blacksquare \beta_j \& 1-\beta_j @ 1-\alpha_j \& \alpha_j \ )$



EM Algorithm (I)

- Conditional Log-Likelihood (iid assumptions):

$$Log\ p\boldsymbol{D\theta},\boldsymbol{x} = \sum i=1\uparrow n \log(a\downarrow i\ p\downarrow i + b\downarrow i\ (1-p\downarrow i))$$

With $p\downarrow i$ = σ(w^t. $x\downarrow i$)

$$a\downarrow i = \prod j=1\uparrow R \alpha\downarrow j\downarrow\uparrow y\downarrow i\uparrow j\ (1-\alpha\downarrow j)\uparrow(1-y\downarrow i\uparrow j)$$

$$b\downarrow i = \prod j=1\uparrow R \beta\downarrow j\downarrow\uparrow(1-y\downarrow i\uparrow j)\ (1-\beta\downarrow j)\uparrow y\uparrow j$$

Find ML estimator of $\boldsymbol{\theta}$=[**α,β,w**] as argmax{$Log\ p\boldsymbol{D\theta},\boldsymbol{x}$ }

→Efficiently solved by EM algo

- E-step: fixed $\boldsymbol{\theta}$ → expectation of $y\downarrow i\uparrow \triangleq \mu\downarrow i$
- M-step: fix $\mu\downarrow i$, and estimate $\boldsymbol{\theta}$

EM Algorithm (II)

- E-step: $\mu_i = p(y_i = 1 | y_i^1, \ldots, y_i^R, x_i, \boldsymbol{\theta})$
$= a_i p_i / a_i p_i + b_i (1 - p_i)$

- **M-step:** find that maximizes
$\sum_{i=1}^{N} \mu_i \log(a_i p_i) + (1 - \mu_i).\log(b_i (1 - p_i))$

➔ $\alpha_j = \sum_{i}^{N} \mu_i y_i^j / \sum_{i}^{N} \mu_i$; $\beta_j = \sum_{i}^{N} (1 - \mu_i)(1 - y_i^j) / \sum_{i}^{N} (1 - \mu_i)$

and w solved by logistic regression with soft labels (or equivalently weighted logistic regression), with soft label = μ_i

Notes

- Could be bayesian, by giving a gaussian prior on w, and beta priors on **α,β**
- → $\alpha_j = a1 + \sum_{i=1}^{N} \mu_i y_{ij} / a1 + a2 + \sum_{i=1}^{N} \mu_i \quad , \beta_j = b1 + \sum_{i=1}^{N} (1-\mu_i)(1-y_{ij}) / b1 + b2 + \sum_{i=1}^{N} (1-\mu_i)$

and L2−regularized soft log regr.

- In practice, start with M-step and μ_i given by average vote.
- Similar developments for regression
- Could be extended to the "no-feature" case (p_i replaced by p_0)
  - E-step: $\mu_i = a_i p_0 / a_i p_0 + b_i (1-p_0)$
  - M-step: estimate **α,β** as before and $p_0 = \sum_{i=1}^{N} \mu_i / N$